DEPARTMENT OF COMPUTER SCIENCES
SCHOOL OF COMPUTING

FLORIDA TECH

APPLIED DISCRETE MATHEMATICS

(VERSION 13/8)

WILLIAM DAVID SHOAFF (JANUARY 9, 2017)

Contents

Figures

Notation and Jargon

Constants

0	Zero: First natural number: $0 = 0$
1	One: First positive integer: $1 = 1$
10	Ten: Base for writing numbers in decimal; not to be confused with two $= (10)_2$
2	Two: Base for writing numbers in binary
γ	Gamma: Euler's constant $\gamma \approx 0.5772\ldots$
$\hat{\varphi}$	Phi-hat: Conjugate of the golden ratio: $\hat{\phi} = \frac{1-\sqrt{5}}{2} \approx -0.618\ldots$
π	Pi: Circumference of circle with diameter 1; area of a circle with radius 1: $\pi \approx 3.142\ldots$
$\sqrt{2}$	Square root of $\sqrt{2} \approx 1.414\ldots$
φ	Phi: golden ratio $\phi = \frac{1+\sqrt{5}}{2} \approx 1.618\ldots$
e	e: Base for natural exponential and logarithm: $e \approx 2.718\ldots$

Functions

10^n	Exponential base 10: 10 to the nth power
2^n	Exponential base 2: 2 to the nth power
$\lvert n \rvert$	Absolute value of n; the magnitude of n
$n^{\underline{m}}$	Falling factorial power: $n^{\underline{m}} = n(n-1)(n-2)\cdots(n-m+2)(n-m+1)$
$\frac{d}{dx}f(x)$, $f'(x)$	Derivative of $f(x)$
$\int f(x)dx$	Indefinite integral of $f(x)$
$\lceil x \rceil$	Ceiling of x; smallest integer greater than or equal to x
(x)	Round of x; the integer closest to x
$\lfloor x \rfloor$	Floor of x; greatest integer less than or equal to x
$\lg(x)$	Binary logarithm: $\lg x = \log_2 x$

$\ln(x)$ Natural logarithm: $\ln x = \log_e x$

$\log(x)$ Common logarithm: $\log x = \log_{10} x$

$\log_b(x)$ Logarithm base b: $\log_b x$

$\max(a, b)$ Maximum value of a and b

$\min(a, b)$ Minimum value of a and b

$\phi()$ Empty function ϕ

$\pi(n)$ Prime counting function: Number of primes less than n

$n^{\overline{m}}$ Rising factorial power: $n^{\overline{m}} = n(n+1)\cdots(n+m-1)$

$\sqrt{x}$ Principle square root of x: $(\sqrt{x})^2 = |x|$

$\varphi(n)$ Euler's totient function

$\{x\}$ Fractional part of x

e^n Natural exponential: e to the nth power

$f(\mathbb{X})$ Range of function $f : \mathbb{X} \to \mathbb{Y}$

$f(n) = O(g(n))$ f is big-O of g; $f(n)$ is of order $g(n)$

$f : \mathbb{X} \to \mathbb{Y}$ Function f maps elements from $\mathbb{X}$ into elements in $\mathbb{Y}$

$f : \mathbb{X} \overset{1:1}{\Longrightarrow} \mathbb{Y}$ f is a one-to-one function from $\mathbb{X}$ into $\mathbb{Y}$

$f : \mathbb{X} \underset{\text{onto}}{\overset{1:1}{\Longrightarrow}} \mathbb{Y}$ f is a one-to-one function from $\mathbb{X}$ onto $\mathbb{Y}$

$f : \mathbb{X} \overset{\text{onto}}{\Longrightarrow} \mathbb{Y}$ f is a function from $\mathbb{X}$ *onto* $\mathbb{Y}$

f^{-1} Inverse of function f

$O(g(n))$ Big O of $g(n)$

$x^0, x^1, x^2, x^3, \ldots, x^n$ Power functions; standard basis for polynomials

Miscellaneous

λ Wavelength of a standing wave; also used to denote the empty string

Hz Hertz or cycles per second

Logic

$(\exists\, x)$ Existential quantifier: "for some x"

$(\exists\, x)(\mathrm{P}(x))$ There exists an x such that $\mathrm{P}(x)$ is `True`

$(\forall\, x)(\mathrm{P}(x))$ For all x $\mathrm{P}(x)$ is `True`

$(\forall\, x)$ Universal quantifier: "for all x"

$\wedge$, &&, $\cdot$ Symbols for logical AND

$\neg$, !, $-$ Symbols used for logical NOT

$\vee$, $\|$, $|$ Symbols for Logical OR

$\equiv$, $\iff$ Logical EQUIVALENT

`False`, F, $\perp = 0$ Symbols used for `False`

$\implies$, $\rightarrow$ Symbols for Logical IMPLICATION (If ... Then)

$\ni$ Such that (archaic)

$\oplus$ Logical Exclusive Or

$\mathrm{P}(x)$ Predicate statement: truth depends on x

P Proposition: Either `True` or `False`

P, Q, R Propositional variables

$\therefore$ Therefore

`True`, T, $\top = 1$ Symbols for `True`

Numbers

$(0, \infty)$ Positive real numbers

$(a_{n-1}a_{n-1}\cdots a_1a_0)_b$ Base b number, $a_k \in \{0, 1, \ldots, b-1\}$; Value is $\sum_{k=0}^{n-1} a_k b^k$

$[0, 1]$ Unit interval; real numbers satisfying $0 \leq x \leq 1$

$\aleph_0$ Aleph naught — cardinality of the natural numbers

$\aleph_1$ Aleph one — cardinality of the real numbers

$\gcd(a, b)$ Greatest common divisor of integers a and b

∞ Infinity

$\mathrm{lcm}(a, b)$ Least common multiple of a and b

$\lfloor a/b \rfloor$ or $a \div b$ Quotient when a is divided by b

$\mathbb{B}$ Bits: $\{0, 1\}$

$\mathbb{D}$ Digits: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$\mathbb{H}$ Hexadecimal numerals: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$

$\mathbb{O}$ Octal numerals: $\{0, 1, 2, 3, 4, 5, 6, 7\}$

$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n$ Stirling's approximation of n factorial

a mod b or a % b Remainder when a is divided by b

$a \perp b$ a and b are relatively prime; $\gcd(a, b) = 1$

m and n Integer variables

M_n Mersenne number: $M_n = 2^n - 1$

p and q Variables that are prime numbers

$successor(n) = n + 1$ Successor of n; Natural number after n

x and y Real variables

Counting

$\binom{n}{k}$, $C(n,k)$, ${}_nC_k$ Binomial coefficient; Combinations of n objects taken k at at time.

$a(n)$ or a_n a general term in a sequence of numbers

$a_{\pi(0)}, a_{\pi(1)}, a_{\pi(2)}, \ldots, a_{\pi(n-1)}$ A permutation of elements $a_0, a_1, a_2, \ldots a_{n-1}$

Relations

$[a]$ Equivalence class of element a

$[k]_m$ Equivalence class of integer k modulo m

$\approx$ Approximately equal

$a \equiv b \pmod{m}$ a is congruent to b modulo m; m divides $a - b$

$a \sim b$ a is related to b by some rule

$d \mid a$ Integer d divides integer a; there exists integer c such that $a = dc$

$d \nmid a$ Integer $d \neq 0$ does not divides integer a

$R \subseteq \mathbb{A} \times \mathbb{B}$ R is a relation from $\mathbb{A}$ to $\mathbb{B}$

Sets

(a, b) An ordered pair

$2^{\mathbb{A}}$ Power set of set $\mathbb{A}$; set of all subsets of $\mathbb{A}$

$\neg\mathbb{A}$ Set complement of $\mathbb{A}$

$|\mathbb{A}|$ Cardinality of a set; count of elements in $\mathbb{A}$

$\emptyset$ Empty set $\emptyset$; set that contains no members (elements)

$\mathbb{A} - \mathbb{B}$ Set difference: $x \in \mathbb{A} - \mathbb{B} \implies (x \in \mathbb{A}) \wedge (x \notin \mathbb{B})$

$\mathbb{A} \cap \mathbb{B}$ Intersection of sets $\mathbb{A}$ and $\mathbb{B}$

$\mathbb{A} \cap \mathbb{B} = \emptyset$ Sets $\mathbb{A}$ and $\mathbb{B}$ are disjoint

$\mathbb{A} \cup \mathbb{B}$ Union of sets $\mathbb{A}$ and $\mathbb{B}$

$\mathbb{A} \oplus \mathbb{B}$ Symmetric difference of set $\mathbb{A}$ and set $\mathbb{B}$

$\mathbb{A} \subset \mathbb{B}$	$\mathbb{A}$ is a proper subset of $\mathbb{B}$
$\mathbb{A} \subseteq \mathbb{B}$	$\mathbb{A}$ is a subset of $\mathbb{B}$
$\mathbb{A} \times \mathbb{B}$	Cartesian product of $\mathbb{A}$ and $\mathbb{B}$
$\mathbb{N}$	Natural numbers: $\{0, 1, 2, 3, 4, 5, 6, \ldots\}$
$\mathbb{P}$	Prime numbers: $\{2, 3, 5, 7, 11, 13, \ldots\}$
$\mathbb{Q}$	Rational numbers: $\{a/b \ : \ a, b \in \mathbb{Z}, b \neq 0\}$
$\mathbb{Q}_{fp}$	Floating point numbers
$\mathbb{R}$	Real numbers; the continuum
$\mathbb{R} - \mathbb{Q}$	Irrational numbers
$\mathbb{R}^2$	Two dimensional Euclidean space
$\mathbb{R}^3$	Three dimensional Euclidean space
$\mathbb{U}$	Universal set
$\mathbb{Z}^+$	Positive integers: $\{1, 2, 3, 4, 5, \ldots\}$
$\mathbb{Z}$	Integer: $\{0, \pm 1, \pm 2, \pm 3, \pm 4, \pm 5, \pm 6, \ldots\}$
$\mathbb{Z}_m$	Modular integers: $\{0, , 1, 2, \ldots, m-1\}$
$x \in \mathbb{A}$	x is a member (element) of set $\mathbb{A}$
$x \notin \mathbb{A}$	x is not a member (element) of set $\mathbb{A}$
$\alpha n + \beta$	n^{th} term in an arithmetic sequence with initial condition β and slope α
$\alpha \rho^n$	n^{th} term in a geometric sequence with initial condition α and ratio ρ
β_n	n^{th} term in busy beaver sequence
$\binom{n}{k}$	Pascal number; n choose k; binomial coefficient
δ_n	Number of divisors of n
$\left[{n \atop k}\right]$	Stirling's number of the first kind
π_n	n^{th} prime number
$\vec{A}$	Alice sequence: $\vec{A} = \langle 1, 1, 1, 1, 1, \ldots \rangle$
$\vec{F}$	Fibonacci sequence: $\vec{F} = \langle 0, 1, 1, 2, 3, \ldots \rangle$
$\vec{G}$	Gauss sequence: $\vec{G} = \langle 0, 1, 2, 3, 4, \ldots \rangle$
$\vec{M}$	Mersenne sequence: $\vec{M} = \langle 0, 1, 3, 7, 15, \ldots \rangle$
$\vec{P}$	Prime sequence: $\vec{P} = \langle 2, 3, 5, 7, 11, \ldots \rangle$

$\vec{T}$ Triangular sequence: $\vec{T} = \langle 0, 0, 1, 3, 6, \ldots \rangle$

$\left\{ {n \atop k} \right\}$ Stirling's number of the second kind

C_n n^{th} Catalan number: $C_n = \frac{1}{n+1}\binom{2n}{n}$

H_n n^{th} Harmonic number: $H_n = \sum_{k=1}^{n} \frac{1}{k}$, $H_0 = 0$

$R_n = 2^{2^n} + 1$ Fermat number

Strings and Machines

$|w|$ Length of string w

λ Empty string λ; String with no characters; also used for wavelength

$\mathbb{L}$ $\mathbb{L} \subseteq \Sigma^*$; A language over the alphabet Σ

Σ Sigma — A finite alphabet

Σ^* Kleene closure of Σ; Set of all strings over Σ

Σ^0 $\{\lambda\}$; The language with only the empty string λ

Σ^n The set of all strings of length n

$a, b, c, \ldots$ Symbols in an alphabet

$M = (Q, \Sigma, \delta, q_0, F)$ A finite automaton

$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ A Turing machine

s, t, u, v, w String variables

$w : v, wv$ Concatenation of strings w and v

w^R Reversal of string w

Summations and Differences

$\nabla f(n)$ Backward difference: $\nabla f(n) = f(n) - f(n-1)$

$\delta f(n)$ Central difference: $\delta f(n) = f(n+1) - f(n-1)$

$\triangle f(n)$ Forward difference of $f(n)$: $\triangle f(n) = f(n+1) - f(n)$

$\prod_{k=0}^{n-1} a_k$ Product of factors a_k for $k = 0$ to $k = (n-1)$

$\sum a_k$ Sum of terms a_k over some index set

$\sum$ Summation symbol

$\sum_0^{-1}$ Empty sum $\sum_0^{-1}$: equal to 0

$\sum_{k=0}^{n-1} a_k$ Sum of terms a_k for $k = 0$ to $k = (n-1)$

$f[n, m]$ Divided difference: $f[n, m] = \frac{f(m) - f(n)}{m - n}$

i, j, k Index variables

Preface

There should be no such thing as boring mathematics.

Edsger Dijkstra, Introduction to his 1987 Mathematical Methods course.

So we confront a mystery. How does it happen that mathematics has remained as it were a blind spot in our culture — alien territory, in which only the elite, the initiate few have managed to entrench themselves?
It's the same old refrain: "Stop, for heaven's sake! I hate math." "Pure torture from the start of school. It's a total mystery how I ever managed to graduate." "A nightmare for me — I have no talent for it, period!" . . . "Mathematical formulas are pure poison. They just turn me off."
Complaints such as these are heard all the time. Thoroughly sensible, educated people express them routinely, with a remarkable blend of defiance and pride.

Hans Magnus Enzensberger

IT'S WEIRD HOW PROUD PEOPLE ARE OF NOT LEARNING MATH WHEN THE SAME ARGUMENTS APPLY TO LEARNING TO PLAY MUSIC, COOK, OR SPEAK A FOREIGN LANGUAGE.

Forgot Algebra

Continuous & Discrete mathematics

Many students enter college prepared, through a series of lessons, to delve deeper into continuous mathematics: calculus, differential equations, real analysis, and onward. These notes are about another perspective: Discrete mathematics, which can provide approximate answers to questions about continuous models, but discrete mathematics can do much more than that, especially as preparation for algorithmic thinking for general computing.

You might liken the relation between continuous and discrete mathematics to the wave-particle duality of modern physics. Duality is an general concept unifying many branches of mathematics, science & engineering: Knowledge of structures in one system can infer knowledge of structures in the other.

My notes are a broad* survey of discrete mathematics. Many topics that could be covered are not. The mind-maps, starting on page 401, illustrate what is covered.

There are several natural entry points into discrete mathematics.

- Logic because "yes" and "no" are fundamental answers.
- Numbers and arithmetic because, well, because most people understand numbers and basic arithmetic on them.
- Sets because reasoning about collections of things is useful and intimately related to logic.
- Functions because computable ones can provide solutions to problems.

* These notes are not very deep and they are not very rigorous either. When they are wrong, I would appreciate a note.

The decision problems: "Is some value x a member of some set $\mathbb{X}$?" has a "yes" or "no" answer. The `True` or `False` answer may be easy or hard to compute: This is a question of *complexity*. It may be that the answer can or cannot be computed: This is a question of *computability*.

I choose to start by giving a high-level overview of computer architecture. Abstract computing machines use logic to define numbers, perform numeric operations, decide problems about collections of things, and design high-level functions for all sorts of purposes.

Turing machines are one model of computation.

The notes are very much a draft. You should check back often to get the latest version.

"'To do" notes like this, point to things that need to be written and re-written.

Thanks

All I've done is collect the ideas of many other people and put them together in a way which, I hope, first-year college students can understand. The ideas are useful in algorithmic reasoning.

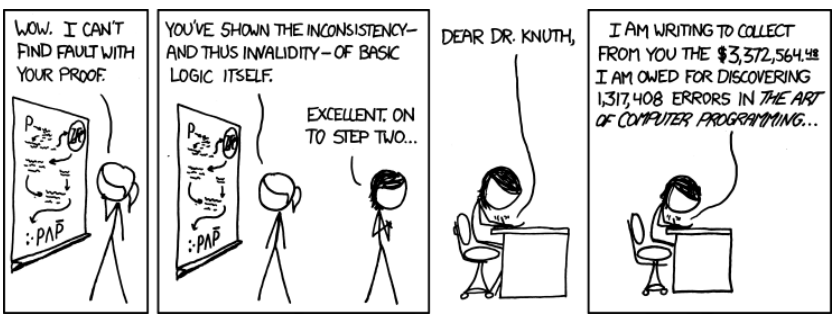
I would like thank everyone who has helped me develop my notes. There are too many to count, let alone thank.

I have to start with Donald Knuth. He is the father of discrete mathematics. His series of books "The Art of Computer Programming" (Knuth, 1997a,b, 1998) provide fascinating, but intimidating, reading.

Civil engineering students sometimes enroll in the Stanford course, but usually quietly withdraw.

Concrete Mathematics is famous enough to have its own Wikipedia page.

Knuth also created TEX because he needed a robust document preparation system. LATEX 2_ε, created by Leslie Lamport, is the "layman's" macro package built on top of TEX.



Applied Math

Barrel Part 1

Then, there is the textbook Concrete Mathematics (Graham et al., 1989) that Knuth wrote with Ronald Graham and Oren Patashnik for a Stanford University class of the same name. It is the best mathematics text I've ever read. Concrete Mathematics is a good supplement for this course and for further study. The inspiration to use marginal notes comes from Concrete Mathematics Some of the side notes in that text are "graffiti" contributed by students. I invite you to suggest graffiti for these notes. Edward Tufte also uses marginal notes in his writing. The series of books by Tufte provide good advice on the visualization of information (Tufte, 1990, 1997, 2001, 2006). I encourage you to look at Tufte's rules for data design. My notes were typeset using a Tufte style for LATEX 2_ε.

"An initial design of the (Wikipedia) logo was created by Paul Stansifer, a 21-year-old Wikipedia user (and son of Dr. Ryan Stansifer), whose entry won a design competition run by the site in 2003."

I want to thank Randall Munroe for publishing the xkcd comic strip.

My thanks go to Jimmy Wales for creating Wikipedia. And, my thanks go to Haskell Curry for the HASKELL programming language. This course is not about computer programming, but it is good to know that mathematical ideas can be expressed in a computer programming language. The converse seems more important: A computer programming language can abstract problems

Like Pascal, Haskell Curry did not invent HASKELL, but the language is named in his honor: A difference is Haskell is a given name while Pascal is a surname.

There are criticisms, of Wikipedia, but in my humble opinion, it is a good resource for learning some discrete mathematics.

and their solutions to mathematical expressions (functions).

Other things

My bias is: To understand discrete mathematics you must read, write, hear, watch, and speak discrete mathematics. I'd like to think you will enjoy working through my notes about discrete mathematics[†].

There are many fine textbooks on discrete mathematics. I've learned from them. I encourage you to look at them too. Someone else may explain a topic in a way that "clicks" with you, while my notes leave you with: Huh?

- A Short Course in Discrete Mathematics, by Edward Bender and Gill Williamson, Dover, 2005. [An introductory text]
- Discrete Mathematics with Applications (4th edition), by Susanna Epp, Brooks/Cole Publishing Company, 2010. [An introductory text]
- Discrete Mathematics and Its Application (7th edition), by Kenneth Rosen, McGraw-Hill, 2011. [An introductory text]
- Discrete Mathematics with Ducks, by Sarah-Marie Belcastro, Taylor & Francis, 2012. [An introductory text]
- Discrete Mathematics Using a Computer (2nd edition), by John O'Donnell, Cordelia Hall, and Rex Page, Springer-Verlag, 2006. [Uses HASKELL to explain discrete mathematics]
- A Logical Approach to Discrete Mathematics, by David Gries and Fred Schneider, Springer-Verlag, 1994. [A more advanced text]
- Discrete Structures With Contemporary Applications, by Alexander Stanoyevitch, Chapman & Hall, 2011. [A more advanced text]

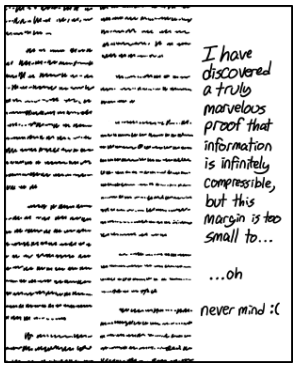
I believe most students taking this course have little experience writing mathematics[‡]. Or, perhaps more correctly, pedantic third grade rules have been enforced in math class, and few students have experience interweaving English (or some other natural language) and mathematics. I encourage you to keep a notebook for this course.

Here's a link to the notebook Issac Newton used while an undergraduate at Trinity College, Cambridge. Perhaps (a few) years from now your notebook will be studied too.

My handwriting is awful. I blame my third grade teacher: Miss Be(a)vis. I write with my left hand. She knew I should use my right hand, but I, being Butt-Head, refused. My students suffer today.

Based on my handwriting and fast lecturing style, I think it would be wise for students to watch and listen to others. Discuss discrete mathematics with your fellow students and others who know or want to know about the subject. There may be good videos on the Internet on topics in discrete mathematics. There are interesting discrete mathematics courses at elite and good universities: Check them out!

Here Munroe's look at the use of margins.



Margin

See "Why Haskell matters"

[†] I do not live in a fairyland.

[‡] I do live in the real world.

I recently found a newspaper article about Miss Bevis' retirement after 40 years of teaching.

A *link* concept: A source maps to a target. You can jump back from the target to the source if there is only one source: This is fundamental idea about functions: If the map (link) from source to target is *not* one-to-one, then you cannot be certain which source to return to from the target.

Within these notes there are many links to external sources. They are typeset in navy blue. These links provide with additional information about a topic.

There are also internal links within the notes. Internal links are typeset in cyan. Let me know if the difference between navy blue and cyan is too subtle. Links to the bibliography are typeset in (Florida Tech) crimson.

Need or want extra respect? Point me to good videos that explain topics covered in my course.

Don't fail in silence!, Dr. Richard Ford's advice to new students, The Florida Tech Crimson, Fall 2011, Issue 2.

Distributed practice also has high utility: Don't cram.

The notes contain many quizzes. They occur after a short topic has been presented. The short quizzes are not graded. You will be asked to complete these quizzes during class. Your classmates may be confused. *Please* ask questions for them. There are a few longer exams. They are templates for proctored, graded exams. *Please* complete these outside of class, as if in a test situation. Cognitive scientist report that frequent practice quizzes have high utility in learning.

Each chapter contains a set of homework questions labeled with a icon. Most often these questions require sustained thinking, and are diagnostic of `True` mastery of a topic. A spectrum of problems from simple to advanced to unsolved or unsolvable are provided.

It would be smart to organize exercises by type.

Speaking mathematics improves learning. You are encouraged to use your professor's and teaching assistant's office hours to talk about discrete math. The "office hours" scene from The PhD Movie! may provide tips on how to prepare for these interactions.

Napolean Dynamite: Well, nobody's going to go out with *me*!

Pedro: Have you asked anybody yet?

Napoleon Dynamite: No, but who would? I don't even have any good skills.

Pedro: What do you mean?

Napoleon Dynamite: You know, like nunchuk skills, bow hunting skills, computer hacking skills. . . Girls only want boyfriends who have great skills.

The PhD Movie!:Office Hours

Finally, there is a web site for the class where you will find a course syllabus and other useful resources.

Are you ready?

Modern learning theory (Fuson et al., 2005) suggests that students fail to learn mathematics (and other subjects) for three primary reasons:

1. Mistaken preconceptions — The "I can't do math!" conjecture: If you have mistaken preconceptions, give them up.

2. Failure to link concepts and processes — The Napolean Dynamite maxim: You need good skills. Most people must practice to develop their skills.

3. Lack of meta-cognition — The Descartes-Schrödinger's cat conundrum: I don't think, therefore I am and I am not. I hope you will think about these notes because you find them interesting and useful.

Pass a Quiz: Preliminary quiz

Take a quiz on page 305 to check your understanding. You can return to here from the quiz.

0. Abstracting a computer

People who are more than casually interested in computers should have at least some idea of what the underlying hardware is like. Otherwise the programs they write will be pretty weird.

Donald Knuth, The Art of Computer Programming, Volume 1, Fascicle 1: MMIX – A RISC Computer for the New Millennium

A Simple Computer

A computer inputs data, performs some analysis or work on it, and produces output information. Figure 0 abstracts the basic components of a computer.

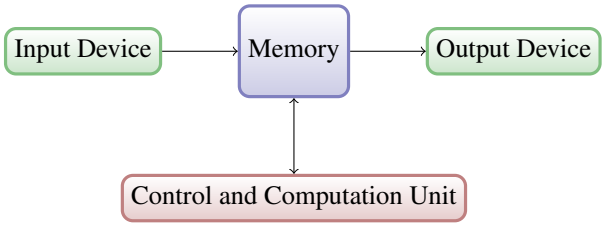


Figure 0: Major computing components

Memory is used to store input, output, and scratch work needed by the control and computation unit to transform input into output. Given a computation, an algorithm, some basic questions are:

- How many control and computation "steps" are taken (as a function of input size)?
- How often is memory read or written?
- How much memory is used?
- How much data is read in or written out?

Binary, Decimal and Hexadecimal

Arithmetic will be reviewed in the next section of notes. Here the idea is to become familiar with notations used in computing.

There are three common bases for writing numbers are decimal, binary, and hexadecimal.

- Decimal— The *lingua franca* for everyday discourse about numbers. Decimal numbers are written using digits from the alphabet

$$\boxed{\mathbb{D} = \{0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9\}} \tag{1}$$

  Using n digits, 10^n decimal numbers can be written. For instance,

Decimal string length → count of numbers

Digits	Range	Different names
1	$\langle 0,\ 1, \ldots,\ 9\rangle$	10
2	$\langle 0,\ 1, \ldots,\ 99\rangle$	100
3	$\langle 0,\ 1, \ldots,\ 999\rangle$	1000
n	$\langle 0,\ 1, \ldots,\ 10^n - 1\rangle$	10^n

- Binary: Physical machines that implement binary arithmetic can be (easily) built. Because of this, people study binary number systems.

  Binary numbers are written using bits from the alphabet

$$\boxed{\mathbb{B} = \{0,\ 1\}} \tag{2}$$

  Using n bits, 2^n binary numbers can be written. For instance,

Binary string length → count of numbers

Bits	Range	Different names
1	$\langle 0,\ 1\rangle$	2
2	$\langle 0,\ 1,\ 2,\ 3\rangle$	4
3	$\langle 0,\ 1, \ldots,\ 7\rangle$	8
n	$\langle 0,\ 1, \ldots,\ 2^n - 1\rangle$	2^n

  Since $2^n \approx 10^{3n/10}$, binary words are about $10/3$ times longer than their decimal equivalent. For instance,

  - The binary number $(1100\ 1010\ 1111\ 1110)_2$ is 16 bits wide: About three and one-third times longer than its 5-digit decimal equivalent 51966.
  - The 7 bits number $(100\ 1001)_2$, when written as decimal 73 takes about three tenths the space: Two digits is about 0.3 of 7.

- Hexadecimal: To decrease the length of binary words, hexadecimal notation is used.

  Hexadecimal numbers are written using hexadecimal numerals from the alphabet

$$\mathbb{H} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\} \tag{3}$$

  Using n hexits, 16^n hexadecimal words can be written. For instance,

Hexadecimal string length → count of numbers

Hexits	Range	Different names
1	$\langle 0, 1, \ldots, 15\rangle$	16
2	$\langle 0, 1, \ldots, 255\rangle$	256
3	$\langle 0, 1, \ldots, 4095\rangle$	16^3
n	$\langle 0, 1, \ldots, 16^n - 1\rangle$	16^n

  Since $2^n = (16)^{n/4}$ binary words are about four times longer than their hexadecimal equivalent. For instance,

  - $(CAFE)_{16}$ is four times shorter than its binary representation: $(1100\ 1010\ 1111\ 1110)_2$.
  - The 6 bits number $(10\ 1010)_2$ requires 2 hexadecimal numberals to write it as $(2A)_{16} = 2 \cdot 16 + 10 = 42$. Two is about one-fourth of six.

Memory

Physical memory is limited: Both in *how many* things it can hold and *the size* of things that can be stored. Let's make several assumptions about our computer's memory.

In practice, today, you can assume a computer has words that are 32 or 64 bits wide. What is the range of natural numbers that can be represented for both of these widths?

An abstract computer's memory is capable to store an unlimited, but countable number of things, and the size of things stored is unlimited, but countable.

- Pretend random access memory is organized into words that are one-byte wide. That is, a computer word w stores 8 bits: An eight-bit byte is a word w: A sequence of 8 bits.

$$w = b_0 b_1 \cdots b_6 b_7$$

b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	A word

  I've chosen to use little-endian to store bits. The least significant bit b_0 is stored first, on the left, at a small address. The most significant bit b_7 is stored last, on the right, at a high address.

  If the word is a binary number then its value is

$$b_0 2^0 + b_1 2^1 + b_2 2^2 + b_3 2^3 + b_4 2^4 + b_5 2^5 + b_6 2^6 + b_7 2^7$$

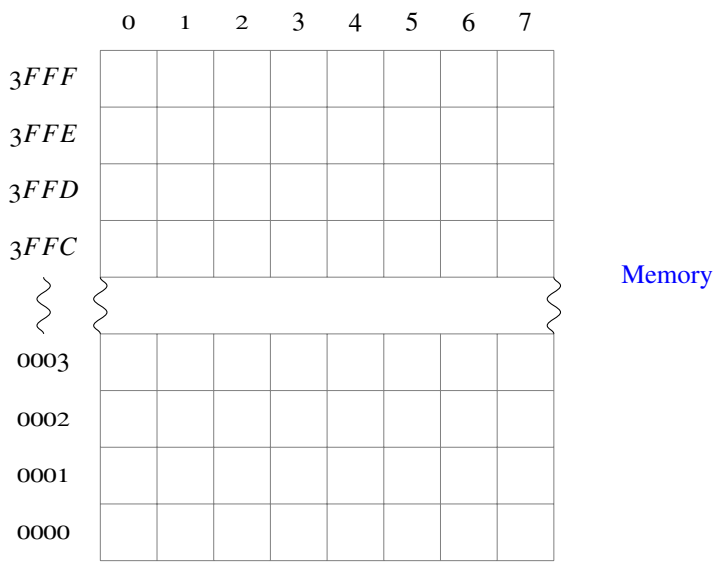


Figure 1: The memory buffer

- Let's suppose there are $N = 2^{14} = (4000)_{16}$ words in primary memory. Each word is one byte (8 bits wide).

  The computer's hardware does not support it, but let's assume memory is bit-addressable: That is, any bit in memory can be read or written by specifying an address (i, j) where $0 \leq i < 2^{14}$ and $0 \leq j < 2^3$.

- Memory can be partitioned into segments: System, instruction, and data segments, but these are details for a computer organization class.

The Control and Computational Units

The central processing unit (CPU) of a computer has a control unit and a computational unit. Control of our machine is fairly simple: It repetitively performs, on clock-ticks, five basic steps.

1. Fetch an instruction (IF)
2. Decode the instruction (ID)
3. Execute the instruction (EX)
4. Access memory (AM)
5. Write back the computed data to the register file (WB)

This establishes an instruction pipeline where, once full, one instruction is completed on each clock-tick.

IF	ID	EX	AM	WB	IF	ID	EX	AM
	IF	ID	EX	AM	WB	IF	ID	EX
		IF	ID	EX	AM	WB	IF	ID
			IF	ID	EX	AM	WB	IF
				IF	ID	EX	AM	WB

Time $\rightarrow$

Of course, there are many things that can *stall* a pipeline.

Control Unit

The central processing unit contains a number of registers that hold control and computational information. The *control unit* contains:

The Intel 8008 served to model some of the features of our computer.

- A 14-bit *program counter* (PC), allowing any word in memory to be addressed.
- A *address register* (AR) that holds the location where data will be stored or fetched. Assume the AR is 2-bytes wide. Although only 14 bits are needed to address memory, the extra two bits can be used to address input and output devices.
- A *call stack* of 8 registers that contain addresses to other active processes running on the machine.
- A 4-bit *status register*: Carry, Parity, Zero, and Sign.
  - The carry bit is set (to 1) when the result of an arithmetic operation resulted in a carry-out or borrow-in of the most significant bit.
  - The parity bit is set (to 1) when the result of the last operation has an even number of 1's
  - The zero bit is set if the result of the last operation was 0.
  - The sign bit is set if the last operation set the most significant bit of the result.

A PC is also called an *instruction pointer* (IP). holds the memory address for the next instruction to execute. Without additional logic, this limits the size of a program to 2^{14} instructions.

Can you deduce the widths of registers in the call stack?

Computational Unit

The computational unit contains other registers.

- An 8-bit *accumulator* (ACC) where the result of the instruction execution is stored.
- An 8-bit *memory-data register* (MDR) that holds data to be stored-in or fetched-from memory in the access memory stage.
- Seven general purpose registers to hold (scratch) data useful for computation and control.

Our computer is programmed by writing instructions that command it to take certain steps.

Instruction set

There are several instructions that can be used to develop programs for our computer. Here are a few:

Assembly instructions

Instruction	Description
HLT	Halt the machine
IN	Input data from a port
OUT	Output data to a port
JMP	Unconditionally jump
JNC	Jump if carry $= 0$
JNZ	Jump if result $\neq 0$
JZ	Jump if result $= 0$
CAL	Call a subroutine
RET	Return from a subroutine
MOV d, M	Load d with content of memory M
ADD s	Add s to ACC
ADD M	Add from memory M to ACC
SUB s	Subtract s from ACC
ANA s	Logical AND of s and ACC to ACC
XRA s	Logical XOR of s and ACC to ACC
ORA s	Logical OR of s and ACC to ACC
CMP s	Compare s with ACC, set flags
NOP	No operation

But this is way too far afield from discrete mathematics. Let's just say a computer can be built and programmed. The details are left to others.

Pass a Quiz: Computer basics

Take a quiz on page 306 to check your understanding. You can return to here from the quiz.

What next?

Now that we have a naive understanding of a computer and its organization, let's first recall what we know about arithmetic and then study how logic can be used to perform arithmetic by controlling the flow of instructions that need to be executed.

Homework Questions

Use your time outside of class to solve these problems.

A general idea to take away from the first three questions is: k things can be put in n places with repetition in k^n ways.

1 Our simple computer, described above, has 8-bit words, a memory with 2^{14} words, a 14 bit program counter, and a 16 bit address register. These numbers are parameters that can be changed. Let

- m be the number of bits in a word
- M be the number of words of memory

- p be the width, in bits, of the program counter
- a be the width, in bits, of the address register

1.1 How many different words of length m can be written?

1.2 How many words can be stored in memory?

1.3 How many instructions can be stored in a program?

1.4 How large must a be to address M words in memory?

2 Pretend our computer is based on decimal notation instead of binary. Answer the four problems from question (1) using digits instead of bits.

3 Pretend our computer is hexadecimal based? How would your answers to the four problems from question (1) change?

4 You quickly realize that 8-bit words are not sufficient for most computations: One byte provides only a small range of numbers. Pretend you want to write a natural number n using the binary alphabet.

4.1 Use the fact that $2^m - 1$ is the largest natural number that can be written using m-bits to find a function that computes ℓ, the number of bits (word length) necessary to write n in binary notation..

4.2 Use your answer to compute the number of bits needed to write: 37, 73, and 237.

5 Next, pretend you want to write n using the hexadecimal alphabets. How many hexadecimal numberals are necessary for 37, 73, and 237? For a general n, how many hexadecimal numberals are necessary?

6 In HASKELL, `Int` is a numeric data type that can represent all integers from -2^{29} to $2^{29} - 1$. How wide, how many bits, must a computer word be to hold any given number in this range? How wide is this in hexadecimal?

The range $[-2^{29}, 2^{29} - 1]$ is the guaranteed minimum range. An implementation of HASKELL may have a larger range of type `Int`.

A useful take-away: If it takes k digits to represent a natural number n, then it will take about $10k/3$ bits to write n. To see this: If it takes k digits to represent a natural number n, then

$$10^{k-1} \leq n < 10^k$$

Replace 10^k with $(10^3)^{k/3}$ and 10^3 with 2^{10}. Then

$$(2^{10})^{(k-1)/3} \leq n < (2^{10})^{k/3}$$

or

$$2^{10(k-1)/3} \leq n < 2^{10k/3}$$

.

7 The Internet Protocol (IP) is used to route traffic on the Internet. IPv4 is being replaced by IPv6. One reason for this is the size of the IPv4 address space.

7.1 IPv4 uses 32-bit (4-byte) addresses. What is the size of its address space? Write your answer in binary and hexadecimal, and approximate the size in decimal. Is this address space large enough in today's world?

7.2 IPv6 uses 128-bit (16-byte) addresses. What is the size of this address space? Write your answer in binary and hexadecimal, and approximate the size in decimal. Do you think the size of this address space will remain large enough for a reasonable long time?

8 Factorials quickly overflow a computer word: $6! = 720$ is already too large for an 8-bit word. What is the largest factorial that can be stored in 16, 32, 64, and 128 bit words?

1. *Arithmetic*

By arithmetic, I mean the study of the traditional operations: Addition, subtraction, multiplication, and division on some underlying set $\mathbb{X}$ of numbers. Changing $\mathbb{X}$ can change whether or not an operation can be defined at all. When an operation cannot be defined on a set, for instance "take 9 from 8 and return a natural numbers," mathematicians enlarge the underlying set so the operation can be performed.

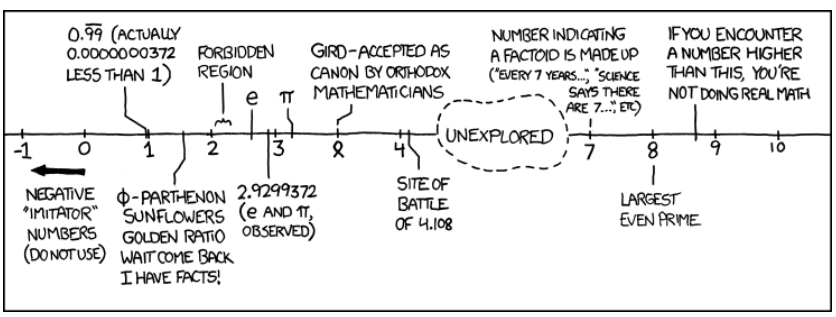
$$\mathbb{N} \xrightarrow{\text{subtraction}} \mathbb{Z} \xrightarrow[\text{division}]{\text{integer exponents}} \mathbb{Q} \xrightarrow[\text{roots}]{\text{rational exponents}} \mathbb{R}$$

Changing $\mathbb{X}$ can also change the algorithms for computing the operations.

Numerical Calculations

Let's discuss how to control a computer to perform numerical computations: Addition, subtraction, multiplication, division (quotients and remainders), exponentiation, and other computable operations and functions.

You know about numbers found along the "number-line."



Number line

Arithmetic will be considered on a hierarchy of numbers.

$$\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R}$$

- Natural numbers $\mathbb{N} = \{0, 1, 2, 3, \ldots, n, \ldots\}$, also called *unsigned*, non-negative, whole numbers, form the basic building blocks for arithmetic. Natural numbers under the operation of addition form a monoid. They also form a monoid under multiplication. That is, these properties are `True`.

Interestingly, although each set $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ is a *proper subset* of the next, these three sets have the same cardinality: The same "count" of elements. On the other hand, the set $\mathbb{R}$ of real numbers is significantly larger.

$$|\mathbb{N}| = |\mathbb{Z}| = |\mathbb{Q}| < |\mathbb{R}|$$

$\mathbb{N}$ is for *natural*, $\mathbb{Z}$ is for *zahlen* the German word for integer, $\mathbb{Q}$ is for *quotient*, which with *remainder* are how rational numbers are identified, and $\mathbb{R}$ is for, well, *real*.

A number has a type: natural, integer, modular, rational, real, complex, etc.

Some people don't include 0 in $\mathbb{N}$. I say "nothing is natural." Of course, "everything is natural" is an interesting cocept too.

Basic arithmetic on the natural numbers

- The natural numbers are closed under addition and multiplication:

  $$\text{If } n,\ m \in \mathbb{N}, \text{ then } n + m \in \mathbb{N} \text{ and } n \cdot m \in \mathbb{N}.$$

  When you add or multiply two natural numbers the sum or product is a natural number.
- Addition and multiplication on the natural numbers obey the associative law: For every $n,\ m,\ p \in \mathbb{N}$,

  $$(n + m) + p = n + (m + p) \quad \text{and} \quad (n \cdot m) \cdot p = n \cdot (m \cdot p)$$

  You can add or multiply in any order, provided the terms or factors remains stationary.
- There are additive and multiplicative identities: Numbers 0 and 1 such that.

  $$0 + n = n + 0 = n \quad \text{and} \quad 1 \cdot n = n \cdot 1 = n$$

To provide an *inverse* for addition, allowing an amount to decrease, the natural numbers are extended to the integers.

- Integers $\mathbb{Z} = \{0,\ \pm 1,\ \pm 2,\ \pm 3, \ldots,\ \pm n,\ \ldots\}$, also called *signed* whole numbers.

  The integers form a group under addition. That is, addition is closed, associative, has an identity element, and each integer has an additive inverse. The existence of additive inverses extend these monoidal properties to form a group.

  People also speak of the *positive integers*

  $$\mathbb{Z}^+ = \{1,\ 2,\ 3, \ldots\}$$

  and the *negative integers*

  $$\mathbb{Z}^- = \{-1,\ -2,\ -3, \ldots\}$$

  An understanding of groups is essential to comprehend modern particle physics.

Integers have additive inverses

Each integer n has an additive inverse $-n \in \mathbb{Z}$ such that

$$n + (-n) = 0$$

The negative of n is $-n$: $-(n) = -n$

The negative of $-n$ is n: $-(-n) = n$

The integers do *not* have multiplicative inverses: For instance, to solve the *linear* equation $5x = 1$ the integers must be extended to the rational numbers. Over the rationals, you can compute $x = 5^{-1} = 1/5$ as the solution of the linear equation $5x = 1$.

- Rational numbers $\mathbb{Q} = \{r = b/m :\ b,\ m \in \mathbb{Z},\ m \neq 0\}$, also called fractions, extend the integers to provide an *inverse* for *non-zero* multiplication: Division, multiplication by fractional parts.

  Let m and b be integers, with $m \neq 0$. The value of x that solves the

linear equation

$$m \cdot x + b = 0 \quad \text{is the rational number } x = m^{-1}(-b) = -\frac{b}{m}.$$

Rational numbers can be written in fractional, fixed-point, or floating-point forms, for instance:

$\frac{7}{8} = 0.875$	fixed point
$\frac{7}{8} = 8.750 \times 10^{-1}$	normalized floating point

Rational arithmetic has many properties

Rational numbers form a field: There are two operations addition $(+)$ and multiplication $(\cdot)$ on the rational numbers $\mathbb{Q}$ with these properties:

- Closure: If r and s are rational numbers, then

$$r + s \in \mathbb{Q}$$
$$r \cdot s \in \mathbb{Q}$$

- Associative For all rational numbers r, s, and t

$$(r + s) + t = r + (s + t)$$
$$(r \cdot s) \cdot t = r \cdot (s \cdot t)$$

- Commutative: For each r and s in $\mathbb{Q}$

$$r + s = s + r$$
$$r \cdot s = s \cdot r$$

- Identity elements: There are rational numbers 0 and 1 such that for every rational number r

$$r + 0 = r$$
$$r \cdot 1 = r$$

- *Inverses:* Each rational r has an additive inverse $-r$, and every non-zero rational $s \neq 0$ has a multiplicative inverse s^{-1}

$$r + (-r) = 0$$
$$s \cdot s^{-1} = 1$$

– Distributive: Multiplication distributes over addition

$$\boxed{r \cdot (s + t) = r \cdot s + r \cdot t}$$

It is often better to *factor* than *distribute*.

$$\boxed{r \cdot s + r \cdot t \to r \cdot (s + t)}$$

- Real numbers $\mathbb{R}$. The structure of the real numbers is beyond the scope of this course, but some real numbers appear in several sections of these notes. The real numbers extend the rational numbers by "filling in the gaps." That is, there are some simple equations that have no solution over the rationals. For instance, there are no rational numbers x that satisfy the equations

$x^2 - 2 = 0$ defines $\pm\sqrt{2}$ and $x^2 - x - 1 = 0$ defines the golden ratio and its conjugate $(1 \pm \sqrt{5})/2$.

$$x^2 - 2 = 0 \quad \text{or} \quad x^2 - x - 1 = 0$$

But here, we don't care too much about this type of problem.

Rotations provide notions of modular arithmetic:

– A rotation by θ followed by a rotation by ϕ is equivalent to one rotation by $\theta + \phi$.
– A rotation by 360° returns the object to its original orientation.
– A rotation by more that 360° can be *reduced* to a rotation between 0 and 360°.

Rotations written in degrees is a vestige of the sexigesimal number system used by the Babylonians. Recall, a rotation by 180° is equivalent to a rotation by π, but this will never be used elsewhere in these notes.

- Modular integers $\mathbb{Z}_m = \{0, 1, 2, 3, \ldots, (m-1)\}$. Some instances of sets of modular integers are given below.

Mod 0:	$\mathbb{Z}_0 = \emptyset$	Mod 3:	$\mathbb{Z}_3 = \{0, 1, 2\}$
Mod 1:	$\mathbb{Z}_1 = \{0\}$	Mod 4:	$\mathbb{Z}_4 = \{0, 1, 2, 3\}$
Mod 2:	$\mathbb{Z}_2 = \{0, 1\}$	Mod 5:	$\mathbb{Z}_5 = \{0, 1, 2, 3, 4\}$

Context often implies which numerical representation (type and notation) is being used for arithmetic. If necessary, explicit syntax can be used to disambiguate the representation. By default, all values are written in decimal, sign-magnitude notation.

Arithmetic on modular integers is similar to, but different from, the arithmetic you learned in third grade. When the modulus $m = p$ is a prime number, the modular integers $\mathbb{Z}_p = \{0, 1, 2, 3, \ldots, (p-1)\}$ form a field. Modular arithmetic is discussed in the notes on modular numbers.

A related concept is the *integers mod m*. Here all of the values in $\mathbb{Z} = \{0, \pm 1, \pm 2, \ldots\}$ are considered: Integers are collected into equivalence classes based on their remainder upon division by m.

The partitions of the integers defined by the mod 5 and mod 7 relations are listed below. The *name* of each equivalence class is the smallest

non-negative integer in set: The (a) modular integer in the tables below.

Equivalence classes mod 5

Integers Mod 5

$[0]_5$	$[1]_5$	$[2]_5$	$[3]_5$	$[4]_5$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
-10	-9	-8	-7	-6
-5	-4	-3	-2	-1
0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Equivalence classes mod 7

Integers mod 7

$[0]_7$	$[1]_7$	$[2]_7$	$[3]_7$	$[4]_7$	$[5]_7$	$[6]_7$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
-14	-13	-12	-11	-10	-9	-8
-7	-6	-5	-4	-3	-2	-1
0	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Congruence mod m is described in more detail in the notes on equivalence relations.

Positional notation

There are several ways to write numbers. You are familiar with decimal notation where digits, written in a sequence, describe a number in terms of powers of 10. For example,

$$3141 = 3 \cdot 10^3 + 1 \cdot 10^2 + 4 \cdot 10^1 + 1 \cdot 10^0$$
$$57721 = 5 \cdot 10^4 + 7 \cdot 10^3 + 7 \cdot 10^2 + 2 \cdot 10^1 + 1 \cdot 10^0$$

This representation of natural numbers is called positional notation. With the use of a *decimal point*, a little syntax, rational numbers can also be written.

$$2.72 = 2 \cdot 10^0 + 7 \cdot 10^{-1} + 2 \cdot 10^{-2} = \frac{272}{100}$$
$$14.142 = 1 \cdot 10^1 + 4 \cdot 10^0 + 1 \cdot 10^{-1} + 4 \cdot 10^{-2} + 2 \cdot 10^{-3} = \frac{14142}{1000}$$

For computing, numbers do not need to be written in decimal notation. In fact, machines that compute in binary are easier to build.

Base alphabets

Natural numbers can be written in various bases: For instance, base 2, 8, 10, and 16, called binary, octal, decimal, and hexadecimal are commonly used. Here are their alphabets.

Binary, Octal, Decimal, Hexadecimal

$\mathbb{B} = \{0, 1\}$
$\mathbb{O} = \{0, 1, 2, 3, 4, 5, 6, 7\}$
$\mathbb{D} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
$\mathbb{H} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$

You can use syntatic sugar to designate the base: Let n be a natural number. To denote the base in which n is written you could write

$$n = (n)_2 = (n)_8 = (n)_{10} = (n)_{16}$$

This is where Lewis Carroll steps in: n is the name of the concept (some natural number), and the various representations

$$n = (n)_2 = (n)_8 = (n)_{10} = (n)_{16}$$

are how you say n in in base 2, 8, 10 16, or whatever. The way you say n will differ from base-to-base.

For demonstrations, numbers will usually be 3-digits or 8-bits long. Three digits can represent values from $0 = 10^0 - 1$ to $999 = 10^3 - 1$. Eight bits, one byte, can represent values from $0 = 2^0 - 1$ to $11111111 = 255 = 2^8 - 1$.

Base 10, decimal is the default number system: The one with which you are familiar and needs no sugar. Here are some examples of numbers written in different bases.

$$73 = (1001001)_2 = (111)_8 = (73)_{10} = (49)_{16}$$
$$37 = (0100101)_2 = (045)_8 = (37)_{10} = (25)_{16}$$

The algorithms that compute conversions between bases are described starting with converting to decimal on page 118, and then inverting decimal to another base on page 121.

Signed numbers

Signed numbers (integers, rational numbers, real numbers) can be written in sign-magnitude, complement, biased, or other notations. Sign-magnitude is how you usually write numbers. For instance, you write

$$-37, -21, +38, +73$$

The plus sign + is commonly dropped and only implied by its absence. In computing, an extra bit of information is needed to disambiguate the sign.

- In binary sign-magnitude and complement notation, a leading (left-most) 0 indicates that a number is positive (or zero). A leading 1 means the number is negative. Complement number systems are discussed in the notes on the machine representation of signed integers. starting on page 142.

- In binary biased notation just the opposite is `True`: If the leading (left-most) bit is 0 the number is negative (or zero). If the leading bit is 1 the number is positive. Biased number systems are discussed in the notes on representation of floating-point exponents. starting on page 152.

In third grade you learned to perform elementary arithmetic on small numbers using sign-magnitude, decimal notation. This knowledge can be used to understand how to build a computer.

3-bit binary numbers	
UnSigned	
000 = 0	100 = 4
001 = 1	101 = 5
010 = 2	110 = 6
011 = 3	111 = 7
Sign-magnitude	
000 = +0	100 = −0
001 = +1	101 = −1
010 = +2	110 = −2
011 = +3	111 = −3
Two's complement	
000 = 0	100 = −4
001 = +1	101 = −3
010 = +2	110 = −2
011 = +3	111 = −1
Biased $b = 3$	
000 = −3	100 = 1
001 = −2	101 = 2
010 = −1	110 = 3
011 = 0	111 = 4

Addition

Addition is straightforward, but does involve carries. Let's add $473 + 756$ place-by-place and record the carry-ins and carry-outs.

Addition $(473 + 756)$

3	2	1	0	place
1	1	0	0	carry-in
	4	7	3	term 1
	7	5	6	term 2
1	2	2	9	sum
	1	1	0	carry-out

Notice carry-outs propagate to the left becoming carry-ins for the next place. Assume a 3 digit accumulator. The sum $473 + 756$ produces an overflow. This is indicated by the last carry-in bit into place 3. This sum cannot be represented using 3 digits.

On the other hand, in ten's complement notation, $(756)_{10c} = -244$ and the sum $473 + 756 = 473 - 244 = 229$ makes perfect sense.

Computing right-to-left, the initial carry-in is 0, and the sum is $0+3+6 = 9$ with a carry-out of 0. Carry-outs propagate to the left becoming carry-ins in the next place.

$c_{\text{in}_0} = 0$, the first carry-in is 0.

$c_{\text{in}_k} = c_{\text{out}_{k-1}}$, $k = 1, 2, 3, \ldots$ the next carry-in is the previous carry-out.

Carry-outs are computed, place-by-place, using the logic

$$c_{\text{out}} = \begin{cases} 0 & \text{if } c_{\text{in}} + t_1 + t_2 < 10 \\ 1 & \text{if } c_{\text{in}} + t_1 + t_2 \geq 10 \end{cases}$$

The sum digit is computed by the familiar rules illustrated below. A carry-out digit (1) is indicated when the sum is 10 or greater, called *overflow*.

Decimal digit $(a+b)$ addition table

							b				
	$+$	0	1	2	3	4	5	6	7	8	9
	0	0	1	2	3	4	5	6	7	8	9
	1	1	2	3	4	5	6	7	8	9	(1)0
	2	2	3	4	5	6	7	8	9	(1)0	(1)1
	3	3	4	5	6	7	8	9	(1)0	(1)1	(1)2
a	4	4	5	6	7	8	9	(1)0	(1)1	(1)2	(1)3
	5	5	6	7	8	9	(1)0	(1)1	(1)2	(1)3	(1)4
	6	6	7	8	9	(1)0	(1)1	(1)2	(1)3	(1)4	(1)5
	7	7	8	9	(1)0	(1)1	(1)2	(1)3	(1)4	(1)5	(1)6
	8	8	9	(1)0	(1)1	(1)2	(1)3	(1)4	(1)5	(1)6	(1)7
	9	9	(1)0	(1)1	(1)2	(1)3	(1)4	(1)5	(1)6	(1)7	(1)8

Summations

Being able to add two numbers is useful. Being able to sum a finite list of numbers is more useful. Consider the following recursive definition.

There is the famous story about Gauss in third grade. He computed $1+2+3+\cdots+100 = 5050$ is a few seconds, confounding his teacher.

1. The sum of terms in the empty list $\langle\rangle$ is 0.

2. The sum of terms in a non-empty list $\langle x, \ldots\rangle$ is x plus the sum of terms in the tail $\langle\ldots\rangle$.

In HASKELL, the `sum` function adds the values in a finite list of numbers, $\langle x_0, x_1, \ldots, x_{n-1}\rangle$.

```
sum :: Num n => [n] -> n  -- sum is a function that maps
                          -- a list [n] of numbers
                          -- to a value n.
sum [] = 0                -- the sum of no values is 0
sum (x:xs) = x + sum xs   -- add the head
                          -- to the sum of the tail
```

Haskell

The mathematical symbol for sum is $\sum$. The sum of n terms in the sequence $\langle x_0, x_1, \ldots, x_{n-1}\rangle$ is denoted by

$$\sum_{k=0}^{n-1} x_k = x_0 + x_1 + x_2 + \cdots + x_{n-1} \quad \text{for } n \geq 0. \tag{4}$$

It is smart to look at boundary (initial, base) conditions. When $n = 0$ the sum on the left is

$$\sum_{k=0}^{-1} x_k$$

There are different ways to interpret this sum. Both can be useful

The empty sum $\sum_0^{-1}$ is equal to 0.

- The number of terms in the sum is n. When $n = 0$ it means there are no terms in the sum. The natural value for adding up nothing is 0. This is the interpretation I use in these notes. The sum of no terms is the empty sum and has value 0.

- Sometimes, terms in some sequences can be sensibly be extended to negative subscripts. But, in this case, you *should* write

$$\sum_{k=-1}^{0} x_k = x_{-1} + x_0$$

Subtraction

Sign-magnitude subtraction is messy. It requires a look-ahead to borrow-in a 10 when subtracting a large digit from a smaller one.

'Can you do Subtraction? Take nine from eight.'
'Nine from eight I can't, you know,' Alice replied very readily: 'but—'
'She can't do Subtraction,' said the White Queen. 'Can you do Division? Divide a loaf by a knife — what's the answer to that?'
Lewis Carroll (Carroll et al., 2000).

Subtraction $(473 - 756)$

3	2	1	0	place
0	1	0	1	borrow-in
	4	7	3	term 1
	7	5	6	term 2
	7	1	7	difference
1	0	1	0	borrow-out

To subtract 6 from 3, borrow 1 from the 7 in next place. This changes: $7 \to 6$ and $3 \to 10 + 3 = 13$. Now subtract 6 from 13 get 7.

Oops! What happened? $473 - 756 = -283$, not 717. *But*, 717 is the ten's complement representation of -283. In 3-digit arithmetic, the sum $717 + 283$ is equal to 1000, which, when truncated to 3 digits, is $000 = 0$.

$$(717)_{10c} = -283 \quad \text{since } (717)_{10c} + (283)_{10c} = (000)_{10c}.$$

Notice, that

$$756 + 717 = 1473$$

Which when truncated to 3 digits is

$$756 + 717 = \cancel{1}473$$

Or, rearranging terms,

$$473 - 756 = -283 = (717)_{10c}$$

The subtraction table below shows the difference of two digits. The lower triangle shows subtraction of a small digit from a larger digit, for instance, since $7 - 3 = 4$ there is a 4 in row 7, column 3.

The upper triangle shows subtraction of a large digit from a smaller digit, for instance

Some flag other than (9) could be used to denote when a borrow is needed. But the use of (9) has some nice properties.

$$3 - 7 \to 13 - 7 = (9)6$$

where (9) is used to indicate the need to borrow-in 10 from the "next place." Notice that

$$13 - 7 = (9)6 \to 13 = (9)6 + 7$$

captures the fact that $6 + 7$ is 3 with a carry of 1. That is,

$$13 = (9)6 + 7 = (9)[(1)3] = (9 + 1) + 3 = 10 + 3$$

Decimal digit $(a - b)$ subtraction table

		b									
	$-$	0	1	2	3	4	5	6	7	8	9
	0	0	(9)9	(9)8	(9)7	(9)6	(9)5	(9)4	(9)3	(9)2	(9)1
	1	1	0	(9)9	(9)8	(9)7	(9)6	(9)5	(9)4	(9)3	(9)2
	2	2	1	0	(9)9	(9)8	(9)7	(9)6	(9)5	(9)4	(9)3
	3	3	2	1	0	(9)9	(9)8	(9)7	(9)6	(9)5	(9)4
a	4	4	3	2	1	0	(9)9	(9)8	(9)7	(9)6	(9)5
	5	5	4	3	2	1	0	(9)9	(9)8	(9)7	(9)6
	6	6	5	4	3	2	1	0	(9)9	(9)8	(9)7
	7	7	6	5	4	3	2	1	0	(9)9	(9)8
	8	8	7	6	5	4	3	2	1	0	(9)9
	9	9	8	7	6	5	4	3	2	1	0

Multiplication

Digit-by-digit multiplication can produce carries from 0 to 8 as shown in the multiplication table below. This makes implementing the control and

computation of multiplies more complex than addition or subtraction.

Decimal digit $(a \cdot b)$ multiplication table

		b									
	$\cdot$	0	1	2	3	4	5	6	7	8	9
	0	0	0	0	0	0	0	0	0	0	0
	1	0	1	2	3	4	5	6	7	8	9
	2	0	2	4	6	8	(1)0	(1)2	(1)4	(1)6	(1)8
	3	0	3	6	9	(1)2	(1)5	(1)8	(2)1	(2)4	(2)7
a	4	0	4	8	(1)2	(1)6	(2)0	(2)4	(2)8	(3)2	(3)6
	5	0	5	(1)0	(1)5	(2)0	(2)5	(3)0	(3)5	(4)0	(4)5
	6	0	6	(1)2	(1)8	(2)4	(3)0	(3)6	(4)2	(4)8	(5)4
	7	0	7	(1)4	(2)1	(2)8	(3)5	(4)2	(4)9	(5)6	(6)3
	8	0	8	(1)6	(2)4	(3)2	(4)0	(4)8	(5)6	(6)4	(7)2
	9	0	9	(1)8	(2)7	(3)6	(4)5	(5)4	(6)3	(7)2	(8)1

Multiplication of natural numbers can be *reduced* to addition. Let n be a natural number. *Assume* the existence of $\text{succ}(n)$, the successor function, that computes $\text{succ}(n) = (n+1)$. Also assume that there is an addition function that computes $\text{add}(a, b) = (a+b)$ for pairs of natural numbers a and b. Then multiplication can be defined by:

- A *base case*: $\text{mult}(0, a) = 0$; zero times a is zero.
- A *rule*: $\text{mult}(\text{succ}(n), a) = \text{add}(\text{mult}(n, a), a)$;

$$(n+1) \cdot a = a \cdot n + a$$

For instance, you can compute:

$$\begin{aligned}
&\text{mult}(0, a) = 0 \\
&\text{mult}(1, a) = \text{add}(\text{mult}(0, a), a) = \text{add}(0, a) = a \\
&\text{mult}(2, a) = \text{add}(\text{mult}(1, a), a) = \text{add}(a, a) = 2a \\
&\text{mult}(3, a) = \text{add}(\text{mult}(2, a), a) = \text{add}(2a, a) = 3a \\
&\qquad\vdots \qquad \vdots
\end{aligned}$$

Division: Quotients and Remainders

Division is a fundamental arithmetic operation. Let $a, m \in \mathbb{Z}$, where $m \neq 0$. You can divide a by m to compute a quotient q and remainder r giving the *quotient-remainder* equation

$$\boxed{a = mq + r} \tag{5}$$

The quotient is $q = \lfloor a/m \rfloor$, the floor of a/m, the largest integer less than or equal to a/m.

The remainder is $r = a \bmod m$, the difference

$$r = a - qm = a - m\left\lfloor \frac{a}{b} \right\rfloor$$

When the divisor $m = 0$, equation (5) remains `True` by setting the remainder r to a. So, following Knuth, et al in (Graham et al., 1989), define

$$\boxed{a \bmod 0 = a}$$

When the divisor $m = 0$ and $r = a$, equation (5) remains `True` for any value of the quotient q. Rather than leave the quotient indeterminate, I believe a choice should be made. I think the correct decisions are

$$\boxed{q = \left\lfloor \frac{a}{0} \right\rfloor = \begin{cases} 1 & \text{if } a = 0 \\ 0 & \text{if } a \neq 0 \end{cases}}$$

Look at the columns in the quotient table: See how quotient repeat n times in column n.

Decimal digit $\lfloor a/m \rfloor$ quotient table

		m									
	$\div$	0	1	2	3	4	5	6	7	8	9
	0	1	0	0	0	0	0	0	0	0	0
	1	0	1	0	0	0	0	0	0	0	0
	2	0	2	1	0	0	0	0	0	0	0
	3	0	3	1	1	0	0	0	0	0	0
a	4	0	4	2	1	1	0	0	0	0	0
	5	0	5	2	1	1	1	0	0	0	0
	6	0	6	3	2	1	1	1	0	0	0
	7	0	7	3	2	1	1	1	1	0	0
	8	0	8	4	2	2	1	1	1	1	0
	9	0	9	4	3	2	1	1	1	1	1

Look at the columns in the remainder table: See how remainders increase by

1, then cycle back to 0.

Decimal digit a mod m remainder table

		m									
	%	0	1	2	3	4	5	6	7	8	9
	0	0	0	0	0	0	0	0	0	0	0
	1	1	0	1	1	1	1	1	1	1	1
	2	2	0	0	2	2	2	2	2	2	2
	3	3	0	1	0	3	3	3	3	3	3
a	4	4	0	0	1	0	4	4	4	4	4
	5	5	0	1	2	1	0	5	5	5	5
	6	6	0	0	0	2	1	0	6	6	6
	7	7	0	1	1	3	2	1	0	7	7
	8	8	0	0	2	0	3	2	1	0	8
	9	9	0	1	0	1	4	3	2	1	0

Theorem 1: Quotient-remainder

Given two integers a and m, with $m \neq 0$, there exist unique integers q and r such that

$$a = mq + r \quad \text{and} \quad 0 \leq r < |m|.$$

Proof: Quotient-Remainder Theorem

Let

$$\mathbb{X} = \{n \geq 0 : n = a - mk \quad \text{where } k \in \mathbb{Z}.\}$$

The set $\mathbb{X}$ is non-empty. To see this notice

- *If $m > 0$, you can choose $k < a/m$ and then $mk < a$ or $a - mk > 0$.*
- *If $m < 0$, you can choose $k > a/m$ and then $mk < a$ or $a - mk > 0$.*

The set $\mathbb{X}$ is bounded below by 0 and therefore it has a least member: Call this least member $r = a - mq$.
Furthermore, it must be that $r < |m|$. Assuming otherwise leads to a contradiction.

- *Assume $r > m > 0$, then $r > r - m > 0$ and*

$$r - m = a - m(q - 1) \in \mathbb{X}$$

But this contradicts that r is the smallest value in $\mathbb{X}$.

- *Assume $m < 0$ and $r > -m$, then $r > r + m > 0$ and*

$$r + m = a - m(q + 1) \in \mathbb{X}$$

And this contradicts that r is the smallest value in $\mathbb{X}$.

The above proof is not constructive: It does not explain how to compute q and r. One way to construct/compute q and r from a and m is to .

```
  (unsigned q, unsigned r) quotient(unsigned a, unsigned m) {
// assert 0 <= m <= a
    if (a == m) return (1, 0);
    q = 0;
    r = a;
    while (r > m) { q = q + 1; r = r - m; }
    return (q, r);
  }
```

Here's a way to construct/compute q and r. In later notes on conversions among number systems it is shown that a can be written in base m notation. That is,

$$\begin{aligned} a &= (r_{n-1} r_{n-2} \cdots r_1 r_0)_m \\ &= r_{n-1} m^{n-1} + r_{n-2} m^{n-2} + \cdots + r_1 m + r_0 \\ &= \sum_{k=0}^{n-1} r_k m^k \end{aligned}$$

for coefficients $r_{n-1}, r_{n-2}, \cdots, r_1, r_0$ where each coefficient is a value in the base m alphabet

$$\{0, 1, \ldots, (m-1)\}$$

The quotient-remainder theorem says every rational number a/b can be written as a "whole" part plus a "fractional" part, where the whole part is an integer and the fractional part non-negative and less than 1. This is called Euclidean division.

$$\frac{a}{b} = q + \frac{r}{|b|} \quad \text{where } 0 \le r < |b|.$$

Consider mod $b = 3$. Starting at $a = 0$, the remainders r increase by 1 until a reaches 3, when the remainders r return to 0.

$$\begin{array}{llll} \cdots & \frac{-3}{3} = -1 + \frac{0}{3} & \frac{0}{3} = 0 + \frac{0}{3} & \frac{3}{3} = 1 + \frac{0}{3} \cdots \\ \cdots & \frac{-2}{3} = -1 + \frac{1}{3} & \frac{1}{3} = 0 + \frac{1}{3} & \frac{4}{3} = 1 + \frac{1}{3} \cdots \\ \cdots & \frac{-1}{3} = -1 + \frac{2}{3} & \frac{2}{3} = 0 + \frac{2}{3} & \frac{5}{3} = 1 + \frac{2}{3} \cdots \end{array}$$

$$\begin{array}{llll}
\cdots & \frac{-3}{-3} = 1 + \frac{0}{3} & \frac{0}{-3} = 0 + \frac{0}{3} & \frac{3}{-3} = -1 + \frac{0}{3} \cdots \\
\cdots & \frac{-2}{-3} = 0 + \frac{2}{3} & \frac{1}{-3} = -1 + \frac{2}{3} & \frac{4}{-3} = -2 + \frac{2}{3} \cdots \\
\cdots & \frac{-1}{-3} = 0 + \frac{1}{3} & \frac{2}{-3} = -1 + \frac{1}{3} & \frac{5}{-3} = -2 + \frac{1}{3} \cdots
\end{array}$$

The remainder function $\text{rem}(n, m)$ can be defined recursively:

- A *base cases*:
  - $\text{rem}(0, m) = 0$: Zero divided by m has remainder 0: $0 = m \cdot 0 + 0$.
  - $\text{rem}(n, 0) = n$: n divided by zero has remainder n: $n = 0 \cdot q + n$.
  - $\text{rem}(n, m) = 0$ if $\text{rem}(n - 1, m) = m - 1$: Remainders cycle mod m.
- A *rule*: $\text{rem}(\text{succ}(n), m) = \text{rem}(n, m) + 1$: Remainders increase by 1 from 0 to $(m - 1)$, when they cycle back to 0.

For instance, you can compute

$\text{rem}(0, 3) = 0$ from the first base case
$\text{rem}(1, 3) = \text{rem}(0, 3) + 1 = 1$ from the rule
$\text{rem}(2, 3) = \text{rem}(1, 3) + 1 = 2$ from the rule
$\text{rem}(3, 3) = 0$ from the third base case
$\text{rem}(4, 3) = \text{rem}(3, 3) + 1 = 1$
$\text{rem}(5, 3) = \text{rem}(4, 3) + 1 = 2$
$\text{rem}(5, 3) = 0$

Now, the quotient function can be defined using recursion.

- A *base cases*: $\text{quot}(0, 0) = 1$ and $\text{quot}(0, m) = 0$ for $m > 0$.
- Rule 1: $\text{quot}(\text{succ}(n), m) = \text{quot}(n, m) + 1$ if $\text{rem}(m, \text{succ}(n)) = 0$.
- Rule 2: $\text{quot}(\text{succ}(n), m) = \text{quot}(n, m)$ if $\text{rem}(m, \text{succ}(n)) \neq 0$.

For instance, you can compute

$\text{quot}(0, 3) = 0$
$\text{quot}(1, 3) = \text{quot}(0, 3) = 0$ since $\text{rem}(3, 1) \neq 0$
$\text{quot}(2, 3) = \text{quot}(1, 3) = 0$ since $\text{rem}(3, 2) \neq 0$
$\text{quot}(3, 3) = \text{quot}(2, 3) + 1 = 1$ since $\text{rem}(3, 3) = 0$

Exponentiation

Computing exponentials is more complex: The exponentiation table below shows that the "carry" is a sequence of digits. The numbers a^b increase

rapidly. However, there is a clear and computable pattern for the least significant digit. For instance, in the row labeled 3 the pattern for the last digit is $\langle 1, 3, 9, 7\rangle$.

Decimal digit (a^b) exponentiation table

		b									
	^	0	1	2	3	4	5	6	7	8	9
	0	1	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	1	1	1	1
	2	1	2	4	8	(1)6	(3)2	(6)4	(12)8	(25)6	(51)2
	3	1	3	9	(2)7	(8)1	(24)3	(72)9	(218)7	(656)1	(1968)3
a	4	1	4	(1)6	(6)4	(25)6	(102)4	(409)6	(1638)4	(6553)6	(26214)4
	5	1	5	(2)5	(12)5	(62)5	(312)5	(1562)5	(7812)5	(39062)5	$(1\cdots 2)5$
	6	1	6	(3)6	(19)6	(21)6	(777)6	(4665)6	(27993)6	$(1\cdots 1)6$	$(1\cdots 9)6$
	7	1	7	(4)9	(34)3	(240)1	(1680)7	(11764)9	(82354)3	$(5\cdots 0)1$	$(4\cdots 0)7$
	8	1	8	(6)4	(51)2	(409)6	(3276)8	(26214)4	$(2\cdots 5)2$	$(1\cdots 1)6$	$(1\cdots 2)8$
	9	1	8	(8)1	(72)9	(240)1	(5904)9	(53144)1	$(4\cdots 6)9$	$(4\cdots 2)1$	$(3\cdots 8)9$

The exponentiation function $\exp(a, b) = a^b$ can be *reduced* to multiplication.

- A *base case*: $\exp(a, 0) = 1$, that is, $a^0 = 1$.
- A *rule*: $\exp(a, \mathrm{succ}(n)) = \mathrm{mult}(a, \exp(a, n))$, that is, $a^{n+1} = a \cdot q^n$.

For instance, you can compute

$$\begin{aligned}
&\exp(a, 0) = 1\\
&\exp(a, 1) = \mathrm{mult}(a, \exp(a, 0)) = \mathrm{mult}(a, 1) = a\\
&\exp(a, 2) = \mathrm{mult}(a, \exp(a, 1)) = \mathrm{mult}(a, a) = a^2\\
&\exp(a, 3) = \mathrm{mult}(a, \exp(a, 2)) = \mathrm{mult}(a, a^2) = a^3\\
&\qquad\vdots \qquad \vdots
\end{aligned}$$

Historically, in the 1920's people began looking for functions that grew so quickly they would be at or beyond the boundary of computation. For instance, the Ackermann function was invented to demonstrate the existence of very complex, but computable functions.

Pass a Quiz: Number basics

Take a quiz on page 306 to check your understanding. You can return to here from the quiz.

Expressions

Finally, let's jump back from arithmetic to mathematics. We need mathematical things to talk about.

An expression is a finite combination of symbols that is *well-formed* using the rules of some *model*. For example, an arithmetic expression on the natural numbers can be defined recursively.

Definition 1: Arithmetic expression

- *Each natural number n is an (arithmetic) expression.*
- *Variables $x, y, z \ldots$ that can stand for a natural number are (arithmetic) expressions.*
- *If e and f are (arithmetic) expressions, then*
  - *(e) is an (arithmetic) expression*
  - *$e + f$ is an (arithmetic) expression*
  - *$e \cdot f$ is (arithmetic) an expression*

Mathematicians invent additional notation to simplify writing arithmetic expressions. For instance,

$$x^k = x \cdot x^{k-1}, \quad \text{for } k > 0 \quad \text{and } x^0 = 1.$$

$$\sum_{k=0}^{n-1} x_k = x_0 + x_i + \cdots + x_{n-1} \quad \text{for } n > 0 \quad \text{and} \sum_{k=0}^{-1} x_k = 0.$$

$$\prod_{k=0}^{n-1} x_k = x_0 \cdot x_i \cdot \cdots \cdot x_{n-1} \quad \text{for } n > 0 \quad \text{and} \prod_{k=0}^{-1} x_k = 1.$$

Boolean expressions are another type of expression.

A bit is either 0 or 1.

Definition 2: Boolean expression

- *Each bit, 0 and 1, is a (Boolean) expression.*
- *Boolean variables P, Q, R ... are (Boolean) expressions.*
- *Let E and G be (Boolean) expressions. Then each of the following is a (Boolean) expressions:*

$$(\mathrm{E}), \quad \neg \mathrm{E}, \quad \mathrm{E} \wedge \mathrm{G}, \quad \mathrm{E} \vee \mathrm{G}$$

Logicians invent additional notation to simplify writing Boolean expres-

sions. For instance,

$\mathrm{P} \Rightarrow \mathrm{Q} = \neg\mathrm{P} \vee \mathrm{Q}$	Implication
$\mathrm{P} \equiv \mathrm{Q} = (\neg\mathrm{P} \wedge \neg\mathrm{Q}) \vee (\mathrm{P} \wedge \mathrm{Q})$	Equivalence
$\mathrm{P} \oplus \mathrm{Q} = (\mathrm{P} \wedge \neg\mathrm{Q}) \vee (\neg\mathrm{P} \wedge \mathrm{Q})$	Exclusive Or
$\bigvee_{k=0}^{n-1} \mathrm{P}_k = \mathrm{P}_0 \vee \mathrm{P}_1 \vee \cdots \vee \mathrm{P}_{n-1}$	Disjunction
$\bigwedge_{k=0}^{n-1} \mathrm{P}_k = \mathrm{P}_0 \wedge \mathrm{P}_1 \wedge \cdots \wedge \mathrm{P}_{n-1}$	Conjunction

Predicate expressions are also possible: They describe expressions over sets of objects. This type of expression involves *quantifiers*: For every $\forall$ and for some $\exists$. *Quantifiers* express that some predicate $\mathrm{P}(x)$ is `True` for all or for some elements x in a set $\mathbb{X}$.

$$(\forall x \in \mathbb{X})(\mathrm{P}(x)) \quad \text{or} \quad (\exists x \in \mathbb{X})(\mathrm{P}(x))$$

Definition 3: Set expression

- *The empty set $\emptyset$ and universal set $\mathbb{U}$ are set expressions.*
- *Set variables $\mathbb{X}$, $\mathbb{Y}$, $\mathbb{W}$ ... are (set) expressions.*
- *If $\mathbb{X}$ and $\mathbb{Y}$ are (set) expressions, then each of the following is a (set) expression:*

$$(\mathbb{X}), \quad \neg\mathbb{X}, \quad \mathbb{X} \cap \mathbb{Y}, \quad \mathbb{X} \cup \mathbb{Y}$$

Equivalences

A fundamental mathematical question is: Are two expression equivalent? Equivalence of expressions mean all possible values on one expression is the same as the corresponding values of the other expression.

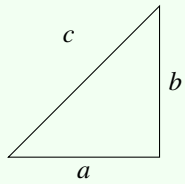
There are several famous equivalences. One is the Pythagorean theorem.

In a mixture of native American and African cultures there once was a warrior who saved an Indian chief. The chief offered one of his daughters as a reward (I do not condone this behavior). The three daughters were sitting on lion, tiger, and hippopotamus hides. The warrior chose the daughter on the hippopotamus hide and the chief asked why? The warrior replied: The squaw on the hippopotamus is equal to the sum of the squaws on the other two hides.

Theorem 2: Pythagorean

The square of the hypotenuse in a right triangle is equal to the sum of the squares of the other two sides.

$$c^2 = a^2 + b^2$$



If some measure other than Euclidean distance is used, the equivalence may not hold. However, the triangle inequality does hold for every metric.

$$|a+b| \leq |a| + |b|$$

Another famous equivalence is Einstein's equation stating the equivalence of energy to mass.

$$\boxed{E = mc^2}$$

There are other fundamental equivalences: Equivalences are always `True`.

- `False` or `True` is always `True`: $\neg \mathrm{P} \vee \mathrm{P} = \texttt{True}$
- `False` and `True` is always `False`: $\neg \mathrm{P} \wedge \mathrm{P} = \texttt{False}$
- `False` implies NOTHING: $(\neg \mathrm{P} \Rightarrow \texttt{True}) \wedge (\neg \mathrm{P} \Rightarrow \texttt{False}) = \texttt{True}$
- If all $x \in \mathbb{X}$ are also in $\mathbb{Y}$ and y is in $\mathbb{X}$, then $y \in \mathbb{Y}$.

Oh the other hand, there expressions can be equal for only certain values: These are not equivalences but *equalities*: In logic they are called contingencies: Statements (predicates) that are `True` for some values and `False` for others.

- The golden polynomial $x^2 - x = 1$ is zero only when x is the golden ratio $\varphi \approx 1.618$, or its conjugate $\hat{\varphi} \approx -0.618$.
- $7x = 3 \bmod 9$ only when $x = 3 \pm 9k$ for $k \in \mathbb{Z}$.

The importance of equivalences is that they allow us to understand complex concepts in more simple ways.

$$\text{(Difficult to understand)} \quad \cos^2\theta + \sin^2\theta = 1 \quad \text{(Easer to understand)}$$

A basic mathematical skill is the ability to show that one expression is equivalent to or equal to another. This is somewhat analogous to writing a computer program: You want to show that a sequence of functions (instructions) produces the desired result, and nothing else. Equivalences are studied in more detail later in these notes.

And then, there is the concept of *inequality*, a whole new realm to study: Inequalities stuides relations such as *smaller–larger*, *inside–outside*, and *before–after*. Placing things in order a *before–after* relationship is studied later too.

What next?

Now that a naive picture of a computer, its organization, and arithmetic has been presented, the next series of notes study how logic can be used to perform arithmetic and other useful operations, how to collect objects into sets, and how to reason about statements (predicates, functions) over collections to control the flow of instructions our computer is executing.

Homework Questions

Use your time outside of class to solve these problems.

1 Evaluate the following expressions.

1.1 2^5

1.2 $2^5 \cdot 2^3$

1.3 $\sum_{k=0}^{5} k$

1.4 $\sum_{k=0}^{-1} 2^k$

2 Pretend you want to write natural number n using the decimal or hexadecimal alphabets. How many numerals are necessary in each case?

3 Pretend you need to name $\jmath$ different things using binary names. Pretend the names will all have the same (fixed) length ℓ. Use the fact that there 2^m different binary strings of length m find a function that computes ℓ,

4 Pretend you want to name $\jmath$ things using (fixed-length) decimal or hexadecimal names. How many digits will be in decimal names? How long will hexadecimal names be?

5 Two and ten are useful bases. A back-of-the-envelope calculation

$$2^{10} = 1024 \approx 1000 = 10^3$$

shows that you can approximate 2^{10} by 10^3, or vice versa.

5.1 What is the absolute error $|2^{10} - 10^3|$ of this approximation?

5.2 What is the relative error $|2^{10} - 10^3| \,/\, |2^{10}|$ in this approximation?

5.3 Use this approximation to write 2^x as a power of 10. Use your result to find an approximation of lg 10, the log base 2 of 10.

5.4 Is the approximation

$$2^{20} = 1048576 \approx 1000000 = 10^6$$

better or worse with respect to absolute and relative error?

The *natural* logarithm function is $\ln x$, the logarithm base e of x, where $e \approx 2.718\cdots$. The *common* logarithm function is $\log x$, the logarithm base 10 of x.

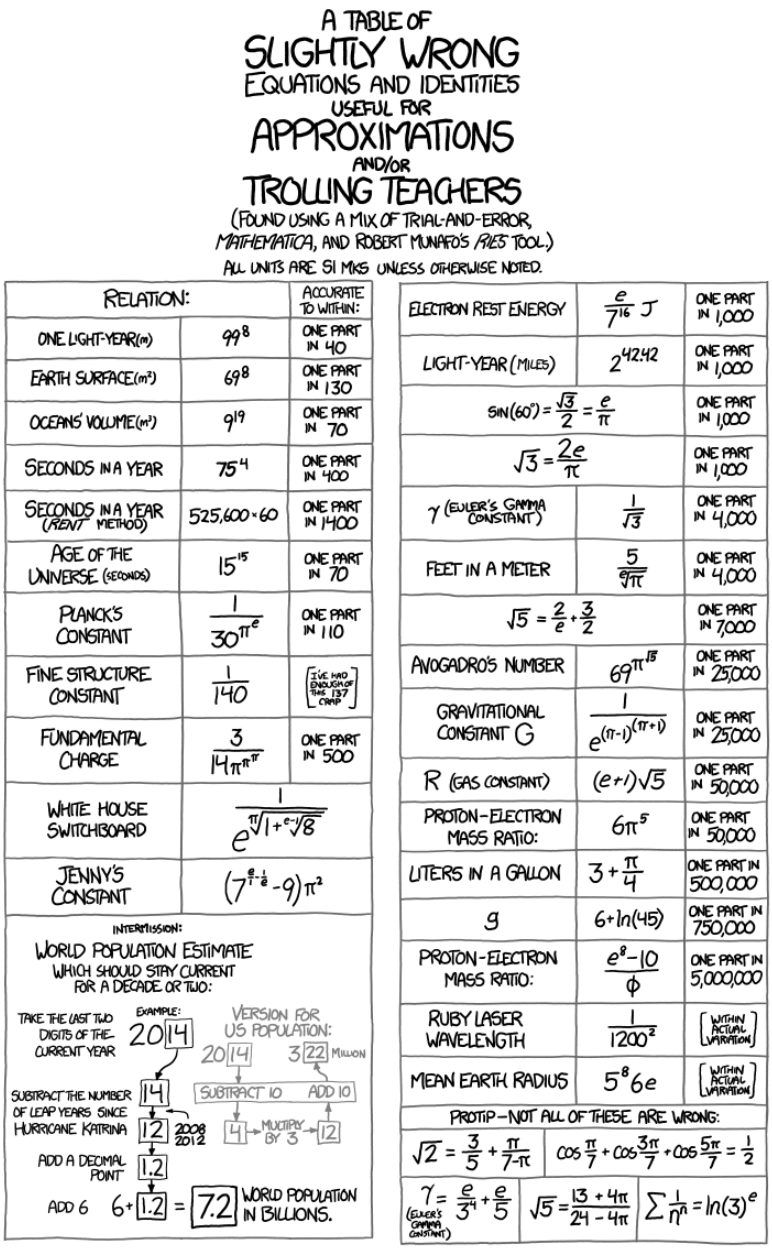
6 Use a calculating tool to approximate $\ln 2$ and $\log 2$. Conclude that their sum $\ln 2 + \log 2$ is close to 1? How close is the approximation?

7 Is exponentiation associative? That is, is the equation

$$a^{(b^c)} = \left(a^b\right)^c$$

`True` for arbitrary (integer) values of a, b, and c? If exponentiation is not associative, what is the accepted *precedence order* for evaluation a^{b^c}: top-down or bottom-up?

8 The xkcd comic below lists several approximations.



Approximations

For instance one light-year is about $9.460730472580,800 \times 10^{15}$ meters, while $99^8 = 9.227446944279201 \times 10^{15}$. The absolute error is the difference

$$(9.460730472580800 - 9.227446944279201) \times 10^{15} = 0.233283528301599 \times 10^{15}$$

While the relative error is only

$$\frac{0.233283528301599}{9.460730472580800} = 0.0246580883979 \approx \frac{1}{40}$$

The point is: Although two numbers may not be what we think of as *close*, they can be *nearby* relative to their magnitudes. Verify the other claims of accuracy in Munroe's comic. Well! That's too big of a request. Choose and work a few of the claims to be certain you understand the ideas.

2. *Logic: For computation*

Wikipedia has a nice list of logic symbols. For each symbol in the list, its name, an explanation, examples, and the symbol's Unicode, HTML, and LaTeX 2_ε values are given.

"I couldn't afford to learn it," said the Mock Turtle with a sigh. "I only took the regular course."
"What was that?" inquired Alice.
"Reeling and Writhing, of course, to begin with," the Mock Turtle replied; "and then the different branches of Arithmetic - Ambition, Distraction, Uglification, and Derision."

Lewis Carroll, Alice's Adventures in Wonderland, Chapter IX (Carroll et al., 2000)

Fundamental Boolean logic

There are two Boolean values: `False` and `True`. Zero (0) is a name for `False` and one (1) is a name for `True`. $\mathbb{B} = \{0, 1\} = \{\texttt{False}, \texttt{True}\}$ is the set of bits.

The symbols $\bot$ and $\top$ are also symbols used for `False` and `True`. $\bot$ is called *bottom* and $\top$ is called *top*.

A Boolean variable is a character used to stand for a Boolean value. Common names for Boolean variables are P, Q, and R.

Claude Shannon popularized the bit as a contraction of "binary digit." (Shannon, 1948). Shannon attributed John Tukey for coining the word.

There are three fundamental *operations* on Boolean variables: `Not`, `And`, and `Or`.

`And` $(\wedge)$ and `Or` $(\vee)$ are *pointy*. The notes on sets discuss *intersect* $\cap$ and *union* $\cup$ which are *round*. The point is: Similar notations are used to describe similar concepts in different contexts.

1. `Not` (Negation)

The `Not` operator: $(\mathrm{P}) \mapsto (\neg\mathrm{P})$

The `Not` Operator

Input	Output	Comment
P	$\neg\mathrm{P}$	
0		
1		

$\neg\mathrm{P}$ is `True` only when P is `False`.

2. `And` (Conjunction)

The And operator: $(P, Q) \mapsto (P \wedge Q)$

The And Operator

Input		Output	Comments
P	Q	$P \wedge Q$	
0	0		
0	1		
1	0		
1	1		

$P \wedge Q$ is `True` only when both P and Q are `True`.

3. `Or` (Disjunction)

The Or operator: $(P, Q) \mapsto (P \vee Q)$

The Or Operator

Input		Output	Comments
P	Q	$P \vee Q$	
0	0		
0	1		
1	0		
1	1		

$P \vee Q$ is `True` except when both P and Q are `False`.

Actually, all you need is `Nand`.

I think it is interesting that `Not`, `And`, and `Or` are all you need to implement arithmetic.

There are several other important Boolean operations that are built using `Not`, `And`, and `Or`. Three are `Conditional`, `Exclusive-Or`, and `Equivalence`.

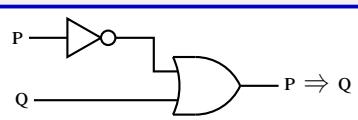
4. `Cond` (Conditional, implies, if . . . , then . . .)

The `Conditional` operator: $(P, Q) \mapsto (P \Rightarrow Q)$

The `Cond` operator: IF P, THEN Q

Input		Output	Comments
P	Q	$P \Rightarrow Q$	
0	0		
0	1		
1	0		
1	1		

$P \Rightarrow Q$ is `True` except when P is `True` and Q is `False`.



$(P \Rightarrow Q) \equiv (\neg P \vee Q)$

I say: If "you score 90+ on the final", then "you will earn an A in the class."

You did not score 90+ and your grade was a B. Was my If... then... statement untrue?

You did not score 90+ and your grade was an A. Was my If... then... statement untrue?

5. `Xor` (Exclusive-Or)

The `Xor` operator: $(P, Q) \mapsto (P \oplus Q)$

The `Xor` Operator

Input		Output	Comments
P	Q	$P \oplus Q$	
0	0		
0	1		
1	0		
1	1		

$P \oplus Q$ is `True` when only one of P or Q is `True`.

6. `Eqv` (Equivalence)

The `Eqv` operator: $(P, Q) \mapsto (P \equiv Q)$

The `Eqv` Operator

Input		Output	Comments
P	Q	$P \equiv Q$	
0	0		
0	1		
1	0		
1	1		

$P \equiv Q$ is `True` when P and Q have the same value.

Precedence order

In arithmetic, the order in which you add, subtract, multiply, divide, exponentiation matters: Different orders gives different results. Order of execution also matters for Boolean operations.

For instance, when $\text{P} = \texttt{False}$, $\text{Q} = \texttt{True}$ and $\text{R} = \texttt{True}$

$$(\text{P} \wedge \text{Q}) \vee \text{R} = \texttt{True}$$
$$\text{P} \wedge (\text{Q} \vee \text{R}) = \texttt{False}$$

$$(a+b)\cdot c \neq a + (b \cdot c)$$

And, precedence defines $a+b\cdot c$ as $a+(b\cdot c)$: Multiplication has *higher* precedence than addition.

From first to last, the precedence of Boolean operations

$$\texttt{Not} \succ \texttt{And} \succ \texttt{Xor} \succ \texttt{Or} \succ \texttt{Cond} \succ \texttt{Eqv}$$

For instance,

$$\text{P} \wedge \text{Q} \vee \text{R} \Rightarrow \neg\text{Q} \wedge \text{R} = ((\text{P} \wedge \text{Q}) \vee \text{R}) \Rightarrow ((\neg\text{Q}) \wedge \text{R})$$

But it is often good to use parenthesis, even if they are not strictly necessary.

Pass a Quiz: Boolean logic basics

Take a quiz on page 307 to check your understanding. You can return to here from the quiz.

Arithmetic

Claude Shannon is often credited with pointing out that Boolean logic can be used to perform arithmetic on binary numbers. A *half-adder* provides a simple example of logic simulating arithmetic: $0+0$ is 0, $0+1$ and $1+0$ are both 1, and $1+1$ is 2, or $(10)_2$ when written in binary.

Syntatic sugar such as $(10)_2$ can be used to clarity that 10 is a binary number, and not the decimal number 10.

More correctly, the carry was 0 for the first three sums. The first three sums did not generate a carry bit, but the fourth sum $1+1$ generated a sum bit $s = 0$ and a carry bit $c_{\text{out}} = 1$.

- carry $=$ (term 1) $\wedge$ (term 2)
- sum $=$ (term 1) $\oplus$ (term 2).

$$\begin{array}{rcrcrcrl} & 0 & & 0 & & 1 & & 1 & \text{term 1} \\ + & 0 & + & 1 & + & 0 & + & 1 & \text{term 2} \\ \hline & 0 & & 1 & & 1 & & 10 & \text{sum} \end{array}$$

It is convenient to write the sums as a *truth table*.

Addition: Half-adder

term 1 + term 2

Input		Output	
term 1	term 2	carry	sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

The carry bit is the conjunction (`And`) of terms 1 and 2, while the sum bit is the exclusive-Or or (`Xor`) of the terms.

A *full-adder* recognizes that carries propagate when terms have more than on place, for example, in decimal.

Note the carry switches from 1 to 0 in the most significant bit. In ten's complement arithmetic this indicates a overflow. The sum 548 is the ten's complement way of writing −452. In ten's complement notation, the sum $173 + 375$ of the positive integers results in a negative value.

Addition $(173 + 375)$

Decimal Addition

$(173 + 375) = 548$

3	2	1	0	place
0	1	0	0	carry in
	1	7	3	term 1
	3	7	5	term 2
	5	4	8	sum
	0	1	0	carry out

$5 + 6 = (101)_2 + (110)_2 = (1011)_2 = 11$

Binary addition $(101 + 110)$

Binary Addition

$(101 + 110 = 1011)$

3	2	1	0	place
1	0	0	0	carry in
	1	0	1	term 1
	1	1	0	term 2
1	0	1	1	sum
	1	0	0	carry out

The first *carry-in* is 0 and each *carry-out* becomes the *carry-in* in the next column.

Full-adder

A full-adder must be able to add two terms (digits, bits) A and B, and a carry-in bit c_{in} to compute a sum bit s and a carry-out bit c_{out}.

The truth table that describes a binary *full-adder* is shown below. Input bits A, B, and c_{in} are summed to produce a sum bit s and a *carry-out* bit c_{out}.

Figure 1: Full-adder truth table

	Input			Output	
Row	A	B	c_{in}	c_{out}	S
1	0	0	0	0	0
2	0	0	1	0	1
3	0	1	0	0	1
4	0	1	1	1	0
5	1	0	0	0	1
6	1	0	1	1	0
7	1	1	0	1	0
8	1	1	1	1	1

There are two *canonical* ways to represent a Boolean function: *disjunctive* and *conjunctive* normal forms. You can construct these normal forms from the input/output behavior of the function.

1. *Disjunctive normal form* (sum of products) Given a truth table: For each row with output 1,

   Form an And-clause of the input variables: Use P when $P = 1$ and $\neg P$ when $P = 0$.

   For instance, referring to full-adder truth table on page 52, the sum bit S is 1 in row 2, 3, 5, and 8.

Sum And-clauses for full-adder

Row	A	B	c_{in}	And-clause
2	0	0	1	$c_2 = \neg A \wedge \neg B \wedge c_{in}$
3	0	1	0	$c_3 = \neg A \wedge B \wedge \neg c_{in}$
5	0	1	1	$c_5 = \neg A \wedge B \wedge c_{in}$
8	1	1	1	$c_8 = A \wedge B \wedge c_{in}$

Now form the disjunction of these clauses.

$$\boxed{S = c_2 \vee c_3 \vee c_5 \vee c_8}$$

This expression for the sum s is `True` on input rows 2, 3, 5 and 8, and it is 0 on all other rows. It can be simplified by factoring and operator

Recognize that

$$(\neg B \wedge C) \vee (B \wedge \neg C)$$

is exclusive-or (`Xor`)

$$B \oplus C$$

and

$$(\neg B \wedge \neg C) \vee (B \wedge C)$$

is equivalence (`Eqv`)

$$B \equiv C$$

which is the negation of $B \oplus C$.

recognition.

$$
\begin{aligned}
\mathrm{S} &= (\neg\mathrm{A}\wedge\neg\mathrm{B}\wedge \mathrm{C_{in}}) \vee (\neg\mathrm{A}\wedge\mathrm{B}\wedge\neg\mathrm{C_{in}}) \vee (\mathrm{A}\wedge\neg\mathrm{B}\wedge\neg\mathrm{C_{in}}) \vee (\mathrm{A}\wedge\mathrm{B}\wedge\mathrm{C_{in}}) \\
&= [\neg\mathrm{A}\wedge((\neg\mathrm{B}\wedge\mathrm{C_{in}})\vee(\mathrm{B}\wedge\neg\mathrm{C_{in}}))]\vee[\mathrm{A}\wedge((\neg\mathrm{B}\wedge\neg\mathrm{C_{in}})\vee(\mathrm{B}\wedge\mathrm{C_{in}}))] \\
&= [\neg\mathrm{A}\wedge(\mathrm{B}\oplus\mathrm{C_{in}})]\vee[\mathrm{A}\wedge\neg(\mathrm{B}\oplus\mathrm{C_{in}})] \\
&= \mathrm{A}\oplus\mathrm{B}\oplus\mathrm{C_{in}}
\end{aligned}
$$

2. *Conjunctive normal form* creates a conjunction of disjunctions (product of sums). For each row with output 0,

   Form an Or-clause of the input variables: Use P when $\mathrm{P} = 0$ and $\neg\mathrm{P}$ when $\mathrm{P} = 1$.

   For example, using rows from the full-adder truth table on page 52, the rows where the carry-out bit is 0 are:

Or-clauses for full-adder carry-out bit

	Input			Output		
Row	A	B	$\mathrm{C_{in}}$	$\mathrm{C_{out}}$	S	Or-clause
1	0	0	0	0	0	$d_1 = \mathrm{A}\vee\mathrm{B}\vee\mathrm{C_{in}}$
2	0	0	1	0	1	$d_2 = \mathrm{A}\vee\mathrm{B}\vee\neg\mathrm{C_{in}}$
3	0	1	0	0	1	$d_3 = \mathrm{A}\vee\neg\mathrm{B}\vee\mathrm{C_{in}}$
5	1	0	0	0	1	$d_5 = \neg\mathrm{A}\vee\mathrm{B}\vee\mathrm{C_{in}}$

   Now form the conjunction of these clauses.

$$\mathrm{C_{out}} = d_1 \wedge d_2 \wedge d_3 \wedge d_5$$

   This expression evaluates to 0 for input rows 1, 2, 3 and 5, and is 1 on all other rows. It can be simplified by factored and operator recognition.

$$
\begin{aligned}
\mathrm{C_{out}} &= (\mathrm{A}\vee\mathrm{B}\vee\mathrm{C_{in}})\wedge(\mathrm{A}\vee\mathrm{B}\vee\neg\mathrm{C_{in}})\wedge(\mathrm{A}\vee\neg\mathrm{B}\vee\mathrm{C_{in}})\wedge(\neg\mathrm{A}\vee\mathrm{B}\vee\mathrm{C_{in}}) \\
&= ((\mathrm{A}\vee\mathrm{B})\vee(\mathrm{C_{in}}\wedge\neg\mathrm{C_{in}}))\wedge(((\mathrm{A}\vee\neg\mathrm{B})\wedge(\neg\mathrm{A}\vee\mathrm{B}))\vee\mathrm{C_{in}}) \\
&= (\mathrm{A}\vee\mathrm{B})\wedge((\mathrm{A}\equiv\mathrm{B})\vee\mathrm{C_{in}})
\end{aligned}
$$

It is possible to chain full-adders together to compute sums of many-bit binary numbers. It should also not as a surprise that subtractions, multiplication, division, exponentiation, and other arithmetic operations can be implemented in logic as well.

Comparisons

Rather than study the implementation of more complex arithmetic, consider comparing two numbers.

There are five basic comparison relations. Each can be reduces to a comparison with 0.

- Less than: $a < b$ if $a - b < 0$
- Less than or equal: $a \le b$ if $a - b \le 0$
- Equal: $a = b$ if $a - b = 0$
- Greater than or equal: $a \ge b$ if $a - b \ge 0$
- Greater than: $a > b$ if $a - b > 0$

Truth tables for $A < B$ and $A \leq B$

Less Than

Input		Output
A	B	$A < B$
0	0	0
0	1	1
1	0	0
1	1	0

$\neg A \wedge B$

Less Than Or Equal

Input		Output
A	B	$A \leq B$
0	0	1
0	1	1
1	0	0
1	1	1

$(\neg A \vee B) \equiv (A \Rightarrow B)$

Truth table for $A = B$

Equality

Input		Output
A	B	$A = B$
0	0	1
0	1	0
1	0	0
1	1	1

$[(\neg A \wedge \neg B) \vee (A \wedge B)] \equiv (A \equiv B)$

Truth tables for $A > B$ and $A \geq B$

Greater Than

Input		Output
A	B	$A > B$
0	0	0
0	1	0
1	0	1
1	1	0

$A \wedge \neg B$

Greater Than Or Equal

Input		Output
A	B	$A \geq B$
0	0	1
0	1	0
1	0	1
1	1	1

$(A \vee \neg B) \equiv (B \Rightarrow A)$

Subtraction

In some sense, there is no subtraction operation: There is only addition of the negative (or complement) of a number.

We are moving up the number hierarchy from natural numbers $\mathbb{N}$ to integers $\mathbb{Z}$.

$$b - a = b + (-a).$$

The representation negative numbers can make addition over integers simple or hard. Three common representations of subtraction $b-a=b+(-a)$ are: Sign-magnitude, complement, and biased notations.

Sign-magnitude notation

Sign-magnitude is the system taught in elementary school and is only slightly covered here.

To compute the difference $b-a$: Subtract the smaller from larger, and change the sign if necessary.

$$b-a=b+(-a)=\begin{cases}+(b-a) & \text{if } a\le b\\ -(a-b) & \text{if } b<a\end{cases}$$

Complement notation

The comparison that occurs in sign-magnitude subtraction can be eliminated using complement notation.

Let's learn about ten's complement first. Pretend our computer has 3-digit words: 000 to 999. Whenever two numbers a and b add to $1000=10^3$, they are ten's complement of each other.

For instance, $(273)_{10c}$ and $(727)_{10c}$ are ten's complement. Their sum is $(273+727)_{10c}=(1000)_{10c}=0$. In this case,

- $(273)_{10c}$ represents 273
- $(727)_{10c}$ represents -273.

In a like manner 1 and 999 are ten's complement with

$$(1)_{10c}=1 \quad \text{and} \quad (999)_{10c}=-1$$

Three-digit ten's complement numbers from $(0)_{10c}$ to $(499)_{10c}$ represent 0 to 499, while ten's complement numbers from $(500)_{10c}$ to $(999)_{10c}$ represent -500 to -1. There is a simple *digit-flipping rule* to negate a ten's complement number:

> *Copy 0's from right-to-left up to the first non-zero digit. Change the first non-zero digit d to $10-d$. Change the remaining digits d on the left to $9-d$.*

For instance, here are some examples showing how to negate ten's complement numbers.

$$\begin{aligned}
-300 &= (700)_{10c} \xrightarrow{\text{negate}} (300)_{10c} = 300\\
-237 &= (763)_{10c} \xrightarrow{\text{negate}} (237)_{10c} = 237\\
375 &= (375)_{10c} \xrightarrow{\text{negate}} (625)_{10c} = -375\\
73 &= (073)_{10c} \xrightarrow{\text{negate}} (937)_{10c} = -73
\end{aligned}$$

You can use syntatic sugar to denote that ten's complement numbers are not your ordinary numbers. For instance, $(375)_{10c}+(625)_{10c}=1000=(0)_{10c}$: $(375)_{10c}$ and $(625)_{10c}$ are 10's complements of each other.

$(375)_{10c}=375$, it's normal interpretation. But $(625)_{10c}-375$. These numbers can be extended to more digits (wider words) by appending 0's or 9's.

$$\begin{aligned}
375 &= (375)_{10c} &&= 0375 = 00375\\
-375 &= (625)_{10c} &&= 9625 = 99625
\end{aligned}$$

Note

$$(1111\ 1111)_2=255$$

and

$$(1\ 0000\ 0000)_2=256.$$

Next, pretend our computer computes with binary numbers and words are 8-bit long: 0000 0000 to 1111 1111. Whenever two numbers a and b add to $(1\ 0000\ 0000)_2 = 256 = 2^8$, they are two's complements of each other.

For instance, $(0110\ 1000)_{2c}$ and $(1001\ 1000)_{2c}$ are two's complements. There sum is

$$(0110\ 1000 + 1001\ 1000)_{2c} = (1\ 0000\ 0000)_{2c} = 0$$

The number $(0110\ 1000)_{2c}$ has it's normal interpretation: 104 in decimal. But, $(1001\ 1000)_{2c}$ represents -104. The unsigned representation $(1001\ 1000)_2$ is 152 in decimal, and $104 + 152 = 256 = 2^8$. For hand computation there is a simple *bit-flipping rule* to negate a two's complement number.

Complement numbers can be extended to more bits (wider words) by appending 0's or 1's.

$$\begin{aligned} 104 &= (01101000)_{2c} \\ &= (0\ 0110\ 1000)_{2c} \\ &= (00\ 0110\ 1000)_{2c} \\ -104 &= (10011000)_{2c} \\ &= (1\ 1001\ 1000)_{2c} \\ &= (11\ 1001\ 1000)_{2c} \end{aligned}$$

> *Copy the bits from right-to-left up to and including the first* 1. *Flip the remaining bits on the left.*

For instance, here are some examples showing how to negate two's complement numbers.

$$\begin{aligned} -30 &= (1110\ 0010)_{2c} \xrightarrow{\text{negate}} (0001\ 1110)_{2c} = 30 \\ -105 &= (1001\ 0111)_{2c} \xrightarrow{\text{negate}} (0110\ 1001)_{2c} = 105 \\ 88 &= (0101\ 1000)_{2c} \xrightarrow{\text{negate}} (1010\ 1000)_{2c} = -88 \\ 73 &= (0100\ 1001)_{2c} \xrightarrow{\text{negate}} (1011\ 0111)_{2c} = -73 \end{aligned}$$

Most teachers say to negate $(a)_{2c}$ Flip each bit: $\neg(a)_{2c}$ and then add 1. Machines can do this easily.

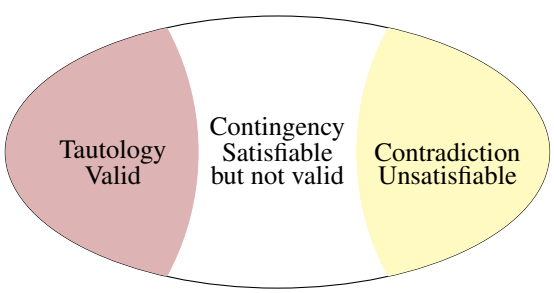
$$b - a = b + (-a) = b + (\neg a + 1)$$

But for hand computation, I think the bit-flipping rule is easier to comprehend.

In general, two n-bit numbers a and b are two's complements of each other if they sum to $a + b = 2^n$. The width, n, of a two's complement number an be widen by prepending 0 to positive and 1 to negative numbers. You'll learn more about how computers represents numbers in the notes on machine numbers.

Tautology, Contradiction & Contingency

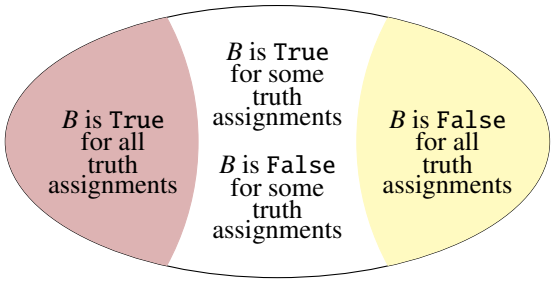
Boolean expressions can be partitioned into three categories, as shown in the diagram below.



If B is a Boolean expression it lies in one and only one of the three categories.

You know, from algebra, that an equation can be:

- Identities: `True` for every assignment of its variables, for instance
$$\frac{n(n-1)}{2} = \frac{(n-1)(n-2)}{2} + (n-1)$$
- Inconsistent: `False` for every assignment of its variables, for instance
$$x = x + 1$$
- Conditional: `True` for some and `False` for assignments of its variables, for instance
$$x^2 - x - 1 = 0$$



A Boolean expression $B(\cdot)$ can be a:

- Tautology: `True` for every possible input. An expression of this type is said to be *valid*. Valid statements can be freely used in *proofs*. When used in proofs, they are called inference rules.

  Here are a few tautologies:

  - `True` (truth)
  - $\mathrm{P} \vee \neg \mathrm{P}$ (law of the excluded middle)
  - $(\mathrm{P} \wedge (\mathrm{P} \Rightarrow \mathrm{Q})) \Rightarrow \mathrm{Q}$ (*Modus ponens*)
  - $(\neg \mathrm{Q} \wedge (\mathrm{P} \Rightarrow \mathrm{Q})) \Rightarrow \neg \mathrm{P}$ (*Modus tollens*)
  - $\neg(\mathrm{P} \vee \mathrm{Q}) \equiv \neg \mathrm{P} \wedge \neg \mathrm{Q}$ (*DeMorgan's law*)
  - $\neg(\mathrm{P} \wedge \mathrm{Q}) \equiv \neg \mathrm{P} \vee \neg \mathrm{Q}$ (*DeMorgan's dual law*)

- Contradiction: `False` for every possible input. An expression of this type is said to be *unsatisfiable*.

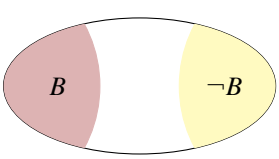
  When You negate a contradiction you get a tautology. For example, from above:

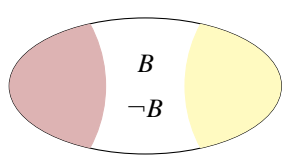
  - `False` (falsity)
  - $\neg \mathrm{P} \wedge \mathrm{P}$ (law of non-contradiction)
  - $(\neg(\mathrm{P} \vee \mathrm{Q}) \wedge (\mathrm{P} \vee \mathrm{Q})$ (I need a name)
  - $\neg((\neg \mathrm{Q} \wedge (\mathrm{P} \Rightarrow \mathrm{Q})) \Rightarrow \neg \mathrm{P})$ (*Not modus tollens*)

Tautologies are satisfiable too.

- Contingency: `True` for some assignments and `False` for others. An expression of this type is called a contingency. A contingency is said to be *satisfiable*. The question for contingencies is: When it it `True` (or `False`)?

Negating a valid Boolean expression produces a contradiction, and vice versa. Negating contingency expression produces another contingency.





Boolean algebra

The Boolean operations ($\neg$, $\vee$ & $\wedge$) form an algebra where these operations play roles similar to *negation*, *addition*, and *multiplication* from arithmetic. The Boolean operations obey laws analogous to those listed on page 27 in the notes on arithmetic. The major difference is that now there are *two distributive laws*.

Boolean algebra properties

- Closure: If P and Q are Boolean expressions, then

$$\begin{aligned} \neg P &\quad \text{is a Boolean expression} \\ P \wedge Q &\quad \text{is a Boolean expression} \\ P \vee Q &\quad \text{is a Boolean expression} \end{aligned}$$

- Associative For all Boolean expressions P, Q, and R

$$\begin{aligned} (P \wedge Q) \wedge R &= P \wedge (Q \wedge R) \\ (P \vee Q) \vee R &= P \vee (Q \vee R) \end{aligned}$$

- Commutative: For each Boolean expression P and Q

$$\begin{aligned} P \wedge Q &= Q \wedge P \\ P \vee Q &= Q \vee P \end{aligned}$$

- Identity elements: There are Boolean expressions `False` and `True` such that for every Boolean expression P

$$\begin{aligned} P \vee \texttt{False} &= P \\ P \wedge \texttt{True} &= P \end{aligned}$$

- *Inverses:* Every Boolean expression P has an negation $\neg P$ such that

$$\begin{aligned} P \vee \neg P &= \texttt{True} \\ P \wedge \neg P &= \texttt{False} \end{aligned}$$

- Distributive:
  1. `Or` distributes over `And`
  2. `And` distributes over `Or`

$$\begin{aligned} P \vee (Q \wedge R) &= (P \vee Q) \wedge (P \vee R) \\ P \wedge (Q \vee R) &= (P \wedge Q) \vee (P \wedge R) \end{aligned}$$

Pass a Quiz: Boolean functions

Take a quiz on page 307 to check your understanding. You can return to here from the quiz.

Homework Questions

Use your time outside of class to solve these problems.

1 Fill in the chart below.

P	Q	$\neg$P	P $\wedge$ Q	P $\vee$ Q	P $\Rightarrow$ Q	P $\oplus$ Q	P $\equiv$ Q
0	0					0	
0	1			1	1		
	0	0					0
			1				

2 Write $P \oplus Q$ using `And`, `Or`, and `Not`.

3 Write $P \equiv Q$ using `And`, `Or`, and `Not`.

4 The Boolean expression $P \Rightarrow Q$ is the conditional statement.

4.1 The converse of $P \Rightarrow Q$ is $Q \Rightarrow P$. Construct a truth table for the converse.

4.2 The inverse of $P \Rightarrow Q$ is $\neg P \Rightarrow \neg Q$. Construct a truth table for the inverse.

4.3 The contrapositive of $P \Rightarrow Q$ is $\neg Q \Rightarrow \neg P$. Construct a truth table for the contrapositive.

4.4 The negation of $P \Rightarrow Q$ is $\neg(P \Rightarrow Q)$. Construct a truth table for the negation.

4.5 Which, if any, of the converse, inverse, contrapositive, or negation is equivalent to the conditional $P \Rightarrow Q$?

5 It is useful to know the number of different ways in which you can assign truth values (`True` or `False`) to Boolean variables. In how many ways can you do this for n variables? If you don't just know, ask yourself what is the answer for small values of n and generalize.

6 What are the 16 Boolean functions on 2 variable. Give each a name.

7 Use the ideas outlined in the notes on the full-adder to show that the carry-out bit c_{out} can be represented by the expression

$$c_{out} = (\neg A \wedge B \wedge c_{in}) \vee (A \wedge \neg B \wedge c_{in}) \vee (A \wedge B \wedge \neg c_{in}) \vee (A \wedge B \wedge c_{in})$$

8 Show that you can write the carry-out function as

$$c_{out} = (A \wedge B) \vee (c_{in} \wedge (A \oplus B))$$

using its disjunctive normal form.

9 A half-subtracter computes the difference of two terms. It can require a borrow-in bit b_{in}. The differences $0 - 0$ and $1 - 1$ are both 0, with no borrows. The difference $1 - 0$ is 1, with no borrow. And, the difference $0 - 1$ is 1, with a borrow-in of 1.

0	0	1	1	term 1
0	1	0	1	term 2
0	11	1	0	difference

Note that in 1-bit arithmetic, $1 + 1 = 0$ with a carry of 1. If $1 + 1 = 0$, then $1 = 0 - 1$ with a borrow-in of 1.

In two's complement notation 11 is the representation of -1.

Construct a truth table for the half-subtracter. What Boolean expressions compute the borrow-in and difference functions?

Note the similarity to decimal arithmetic. For digits $a, b \in \mathbb{D}$, you can subtract $a - b$ without a borrow when $a \geq b$, but when $a < b$ you must borrow from the place on the left. For instance, $9 - 8 = 1$ with no borrow, while $8 - 9 = 9$ using a borrow of 10.

10 There are many useful Boolean expressions. Inference rules are always `True` (a tautology). They can be freely used in proofs. Here are a few. Establish their validity.

"I'm late! I'm late! For a very important date! No time to (explain these laws), goodbye! I'm late! I'm late! I'm late!"

10.1 De Morgan's laws: If not both P and Q are `True`, then one of then is `False`, and vice versa.

$$\neg(P \wedge Q) \equiv \neg P \vee \neg Q$$

10.2 De Morgan's laws: If P or Q is not `True`, then one of $\neg P$ or $\neg Q$ is `False`, and vice versa.

$$\neg(P \vee Q) \equiv \neg P \wedge \neg Q$$

10.3 *Modus ponens*: If P is `True` and P implies Q, then Q is `True`.

$$(P \wedge (P \Rightarrow Q)) \Rightarrow Q$$

Modus ponens is the idea behind direct proofs. Consider the lemma

"if n is even, then n^2 is even."

A direct proof goes like this: Let P denote n is even. Then $n = 2k$ for some integer k. Therefore, $n^2 = 4k^2$, that is n^2 is even.

10.4 *Modus tollens*: If Q is not `True` and P implies Q is `True`, then P is not `True`.

$$(\neg Q \wedge (P \Rightarrow Q)) \Rightarrow \neg P$$

Modus tollens is the idea behind indirect proofs. Consider the lemma

"if n^2 is even, then n is even."

An indirect proof goes like this: Let P denote "n is odd." (do you see the indirection?) Then, $n = (2k + 1)$ for some integer k. Therefore, $n^2 = 4k^2 + 4k + 1$. That is, "n^2 is odd." Let this be Q.

Modus tollens says:

(Not Q (n^2 is even) and P $\Rightarrow$ Q (n is odd implies n^2 is odd)) imply the $\neg$P (n is even).

10.5 Contradiction: If $\neg P$ implies Q and $\neg Q$, then P is `True`.

$$(\neg P \Rightarrow Q) \wedge (\neg P \Rightarrow \neg Q) \equiv P$$

To proof there is no smallest positive rational number, pretend there is one. Call it r. If r is rational and $r > 0$, then $r/2$ is rational and $0 < r/2 < r$. This contradicts the assumption that r is the smallest positive rational. Therefore, $\mathbb{Q}^+$ does not contain its greatest lower bound 0.

10.6 Currying: If P and Q imply R, then P implies that Q implies R, and vice versa.

$$((P \wedge Q) \Rightarrow R) \equiv (P \Rightarrow (Q \Rightarrow R))$$

The idea underlying currying is: The evaluation of function that takes two variables P and Q to compute R can be reduced to evaluating a sequence of two functions: one that takes P as input to produce a function that takes Q as input to compute R

11 I got this one from Click & Clack, the Tappit Brothers: There are three boxes, exactly one of which contains a prize. Each box has a label with a visible statement on it, and exactly one of the statements is `True`.

Gold box label: "The prize is in this box."

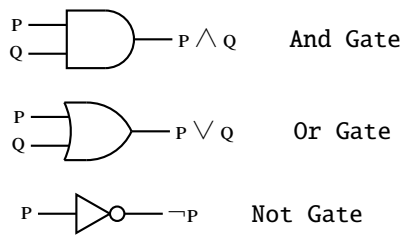
Lead box label: "The prize is not in the gold box."

Silver box label: "The prize is not in this box."

Which box contains the prize?

12 Three logicians walk into a bar. The barkeep asks: Would each of you like a beer? The first logician says "I don't know." The second logician also says "I don't know." The third logician says "Yes, we would." How did the third logician know how to answer the question?

13 Complex computer circuits can be build from three basic gate types, called `And`, `Or`, and `Not`, and drawn as illustrated below.



The inputs P and Q are high (1) or low (0) voltages. Computer engineers are expected to understand how these simple gates can be combined to create complex functions.

13.1 Draw a half-adder gate diagram.

13.2 Draw a full-adder gate diagram.

13.3 Draw a gate diagram for the conditional statement.

14 This problem comes from *To Mock a Mockingbird* (Smullyan, 1985).

Let's play a game. I have two prizes A and B. If you tell me a `True` statement I will give you one of the prizes, but I won't say which one. If you tell me a `False` statement I won't give you either prize.

What statement can you make that guarantees I will give you prize A?

15 To explain Gödel's incompleteness theorem, Smullyan develops many puzzles set on the Island of Knights and Knaves, where everyone is either a knight or a knave. Knights only make `True` statements, while knaves always lie.

Here's a basic puzzle:

The setup: There are two clubs on the island: *Provable* and *Unprovable*. A knight may belong to either club. A knight can say `True` things than can be proven or the knight can say `True` things that have no proof. Knaves cannot be members of either club.

The puzzle: When you visit the island you meet a random islander whose greeting statement allows you to deduce she is a member of *Provable*. What statement could she have made?

16 When you die some of you will be confronted by a riddle and your fate will determine on your answer: You will either go to Heaven or Hell.

Saint Peter guard's Heaven's gates and Cerberus guards Hell, but you won't know who is who because they will be in disguise. Saint Peter always tells the truth and Cerberus always lies.

You can ask them one question and their answer should help you decide which gate leads to Heaven and which is the gaeway to Hell. What question do you ask?

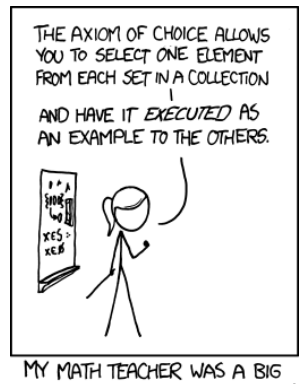
3. Sets: For collections

In other words, general set theory is pretty trivial stuff really, but, if you want to be a mathematician, you need some and here it is; read it, absorb it, and forget it.

Paul Halmos

Later mathematicians will regard set theory as a disease from which one has recovered.

Henri Poincaré



Set Theory

Naive Set Theory

A set is an unordered collection of distinct objects. The objects in a set $\mathbb{X}$ are called *elements* or *members* of $\mathbb{X}$. No element is duplicated in a set: A member of a set is listed once and only once. The order in which the elements are listed does not matter.

A finite set can be described by listing its members in a comma separated list enclosed in curly braces {}. For instance,

- The bits $\mathbb{B} = \{0, 1\}$
- The octits $\mathbb{O} = \{0, 1, 2, 3, 4, 5, 6, 7\}$
- The digits $\mathbb{D} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- The hexits $\mathbb{H} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$
- The modular integers $\mathbb{Z}_m = \{0, 1, 2, \ldots, (m-1)\}$

Simple infinite sets can be described by a clear recurring pattern. For instance,

- The natural numbers $\mathbb{N} = \{0, 1, 2, \ldots\}$
- The integers $\mathbb{Z} = \{0, \pm 1, \pm 2, \ldots\}$

A set $\mathbb{X}$ can be described by *set comprehension*. For instance,

$$\mathbb{B} = \{b : b \text{ is a bit}\} \quad \text{or} \quad \mathbb{EVEN} = \{2n : n \in \mathbb{Z}.\}$$

The symbols $\emptyset$ and $\mathbb{U}$ are symbols used for the set with *no* elements and the set with *every* element.

If x is an element in $\mathbb{X}$ write $x \in \mathbb{X}$. If x is *not* a member of $\mathbb{X}$ write $x \notin \mathbb{X}$.

$x \in \emptyset$ is never `True` $x \notin \emptyset$ is always `True`

$x \in \mathbb{U}$ is always `True` $x \notin \mathbb{U}$ is never `False`

That is, $\mathbb{X}$ can be comprehended by a predicate:

$$\mathbb{X} = \{x : \text{P}(x) \text{ is } \texttt{True}\}$$

where $\text{P}(x)$ is a predicate about variable x. More generally, $\mathbb{X}$ can be comprehended by a function where a predicate is satisfied.

$$\mathbb{X} = \{f(x) : \text{P}(x) \text{ is } \texttt{True}\}$$

The unit interval

$$[0,\ 1] = \{x : 0 \le x \le 1\}$$

The unit circle

$$\{(x,\ y) : x^2 + y^2 = 1\}$$

There are two special sets:

1. The empty set $\emptyset$: The set with no elements, also written $\{\}$.
2. The universal set $\mathbb{U}$: When context demands, the set of all possible things, called the *universal set*, can be described and named. It is also useful for the unversal set to be *computable*. That is, it is good to have an algorithm that *decides* if a given thing x is or is not in the universal set $\mathbb{U}$. But, for instance, the set of real numbers in not computable.

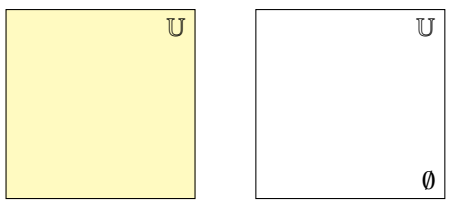
In computing practice, set comprehension requires the function $f(x)$ and predicate $\text{P}(x)$ to be computable. That is, there must be an algorithm that computes $f(x)$ and another algorithm that returns `True` only when $\text{P}(x)$ is `True`.
The most common *universal set* in these notes is $\mathbb{N}$, the set of natural numbers

$$\mathbb{U} = \mathbb{N} = \{0,\ 1,\ 2,\ 3,\ 4,\ 5 \ldots\}$$

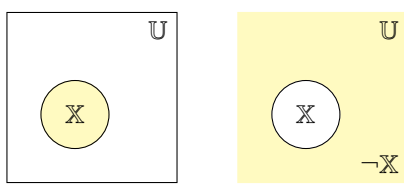
Venn & Euler diagrams

Sets can be visualized by Venn and Euler diagrams. For instance, the universal set $\mathbb{U}$ can be drawn as a shaded square. The empty set $\emptyset$ is then an empty square.

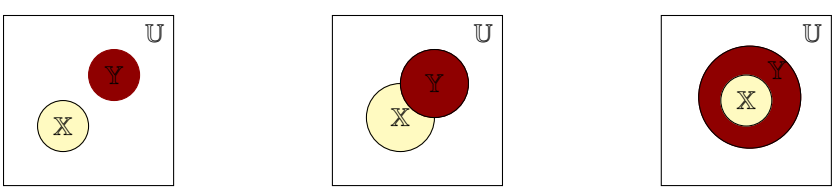
See the Google doodle celebrating John Venn's 180th birthday.



Draw a single circle as set $\mathbb{X}$ inside a square to separate the universal set $\mathbb{U}$ into two region: Inside and outside of $\mathbb{X}$. Shade inside the circle to indicate $\mathbb{X}$. Shade outside the circle to indicate the set complement of $\mathbb{X}$.



Two sets $\mathbb{X}$ and $\mathbb{Y}$ can be drawn in several relationships: Euler diagrams illustrate these.



There are three basis cases on two sets:

1. The two sets are *disjoint*: No members of $\mathbb{X}$ are in $\mathbb{Y}$.

   $$(\forall\ x \in \mathbb{X})(x \notin \mathbb{Y})$$

   Or conversely, no members of $\mathbb{Y}$ are in $\mathbb{X}$.

   $$(\forall\ y \in \mathbb{Y})(y \notin \mathbb{X})$$

2. The two sets *intersect*: Some elements in $\mathbb{X}$ are also in $\mathbb{Y}$.

   $$(\exists\ x \in \mathbb{X})(x \in \mathbb{Y})$$

   Or conversely, when some elements of $\mathbb{Y}$ are in $\mathbb{X}$.

   $$(\exists\ y \in \mathbb{Y})(y \in \mathbb{X})$$

3. Lastly, one set is a *subset* of the other: All elements in $\mathbb{X}$ are in $\mathbb{Y}$

   $$(\forall\ x \in \mathbb{X})(x \in \mathbb{Y})$$

   Or conversely, all elements in $\mathbb{Y}$ are in $\mathbb{X}$

   $$(\forall\ y \in \mathbb{Y})(y \in \mathbb{X})$$

The characteristic function

It is useful to have a function that correctly answers the question: Is x an element of $\mathbb{A}$?

Definition 4: Characteristic Function

The characteristic function of $\mathbb{A}$ is denoted $\chi(x, \mathbb{A})$ and computed by the conditional statement

$$\chi(x, \mathbb{A}) = \begin{cases} \textsc{False} & \text{if } x \notin \mathbb{A} \\ \textsc{True} & \text{if } x \in \mathbb{A} \end{cases}$$

When $\mathbb{X}$ and $\mathbb{Y}$ *intersect*, some element x is in both $\mathbb{X}$ and $\mathbb{Y}$,

$$(\exists\, x)(\chi(x, \mathbb{X}) \wedge \chi(x, \mathbb{Y}) = \texttt{True})$$

The *computational* question is: Given $\mathbb{A}$, is there an algorithm that computes the $\chi(x, \mathbb{A})$? Sometimes you'll be given samples from $\mathbb{A}$ and asked to deduce all of $\mathbb{A}$. For instance, describe how to *compute* elements in these sets:

$$\mathbb{F} = \{0, 1, 1, 2, 3, 5, \ldots\}$$
$$\mathbb{M} = \{0, 1, 3, 7, 15, 31, \ldots\}$$
$$\mathbb{T} = \{0, 0, 1, 3, 6, 10, \ldots\}$$

Some sets are not computablbe. The question: Is $x \in \mathbb{A}$? cannot be answered for some elements x and sets $\mathbb{A}$.

Definition 5: Intersecting sets

Let $\mathbb{X}$ and $\mathbb{Y}$ be sets. $\mathbb{X}$ and $\mathbb{Y}$ intersect if some element x is in both $\mathbb{X}$ and $\mathbb{Y}$.

$$(\exists\, x \in \mathbb{U})((x \in \mathbb{X}) \wedge (x \in \mathbb{Y}))$$

When $\mathbb{X}$ and $\mathbb{Y}$ are *disjoint*, the values $\chi(x, \mathbb{X})$ and $\chi(x, \mathbb{Y})$ cannot both be `True` simultaneously. That is,

$$(\forall\, x)(\chi(x, \mathbb{X}) \wedge \chi(x, \mathbb{Y}) = \texttt{False})$$

Definition 6: Disjoint sets

Let $\mathbb{X}$ and $\mathbb{Y}$ be sets. $\mathbb{X}$ and $\mathbb{Y}$ are disjoint if no element x is in both $\mathbb{X}$ and $\mathbb{Y}$.

$$(\forall\, x \in \mathbb{U})((x \in \mathbb{X}) \wedge (x \in \mathbb{Y}) = \texttt{False})$$

And, when $\mathbb{X}$ is a *subset* of $\mathbb{Y}$, every element in $\mathbb{X}$ also belongs to $\mathbb{Y}$. That is,

$$(\forall\, x)((\chi(x, \mathbb{X}) = \texttt{True}) \Rightarrow (\chi(x, \mathbb{Y}) = \texttt{True}))$$

Or, rewriting the conditional as a disjunction

$$(\forall\, x)((\chi(x, \mathbb{X}) = \texttt{False}) \vee (\chi(x, \mathbb{Y}) = \texttt{True}))$$

Definition 7: Subset

Let $\mathbb{X}$ and $\mathbb{Y}$ be sets. $\mathbb{X}$ is a subset of $\mathbb{Y}$ if every element in $\mathbb{X}$ is also in $\mathbb{Y}$.

$$(\forall\, x \in \mathbb{U})((x \in \mathbb{X}) \Rightarrow (x \in \mathbb{Y}))$$

Pass a Quiz: Basics set concepts

Take a quiz on page 308 to check your understanding. You can return to here from the quiz.

Set operations

There are three fundamental *operations* that can be applied to sets. They are: set complement, intersection and union. They map sets to other sets. These operations can be defined in terms of the Boolean predicate $x \in \mathbb{X}$ which is either `True` or `False`.

1. Set complement (similar to Boolean NOT)

Set complement $(\mathbb{X}) \mapsto (\neg\mathbb{X})$

(Set) Complement $\neg$

Input	Output	Comment
$x \in \mathbb{X}$	$x \in \neg\mathbb{X}$	
0		
1		

$$\neg\mathbb{X} = \mathbb{U} - \mathbb{X}$$

2. Intersection (similar to Boolean AND)

Set Intersection $(\mathbb{X}, \mathbb{Y}) \mapsto (\mathbb{X} \cap \mathbb{Y})$

Intersection $\cap$

Input		Output	Comments
$x \in \mathbb{X}$	$x \in \mathbb{Y}$	$x \in \mathbb{X} \cap \mathbb{Y}$	
0	0		
0	1		
1	0		
1	1		

$x \in \mathbb{X} \cap \mathbb{Y}$ is `True` only when $x \in \mathbb{X}$ and $x \in \mathbb{Y}$ are both `True`.

3. Union (similar to Boolean OR)

Set Union $(\mathbb{X}, \mathbb{Y}) \mapsto (\mathbb{X} \cup \mathbb{Y})$

Union $\cup$

Input		Output	Comments
$x \in \mathbb{X}$	$x \in \mathbb{Y}$	$x \in \mathbb{X} \cup \mathbb{Y}$	
0	0		
0	1		
1	0		
1	1		

$x \in \mathbb{X} \cup \mathbb{Y}$ when $x \in \mathbb{X}$ or $x \in \mathbb{Y}$.

The set operations ($\neg$, $\cup$ & $\cap$) form an algebra where these operations obey exactly the same rules as `Not`, `And`, and `Or` in Boolean logic. The set operation laws are the same to those listed on page 58 in the notes on logic operations.

Boolean algbra properties for sets

- Closure: If $\mathbb{X}$ and $\mathbb{Y}$ are sets, then

$$\begin{aligned} \neg\mathbb{X} &\quad \text{is a set} \\ \mathbb{X} \cap \mathbb{Y} &\quad \text{is a set} \\ \mathbb{X} \cup \mathbb{Y} &\quad \text{is a set} \end{aligned}$$

- Associative For all sets $\mathbb{X}$, $\mathbb{Y}$, and $\mathbb{V}$

$$\begin{aligned} (\mathbb{X} \cap \mathbb{Y}) \cap \mathbb{V} &= \mathbb{X} \cap (\mathbb{Y} \cap \mathbb{V}) \\ (\mathbb{X} \cup \mathbb{Y}) \cup \mathbb{V} &= \mathbb{X} \cup (\mathbb{Y} \cup \mathbb{V}) \end{aligned}$$

- Commutative: For each set $\mathbb{X}$ and $\mathbb{Y}$

$$\begin{aligned} \mathbb{X} \cap \mathbb{Y} &= \mathbb{Y} \cap \mathbb{X} \\ \mathbb{X} \cup \mathbb{Y} &= \mathbb{Y} \cup \mathbb{X} \end{aligned}$$

- Identity elements: There are sets $\emptyset$ and $\mathbb{U}$ such that for every set $\mathbb{X}$

$$\begin{aligned} \mathbb{X} \cup \emptyset &= \mathbb{X} \\ \mathbb{X} \cap \mathbb{U} &= \mathbb{X} \end{aligned}$$

- *Inverses:* Every set $\mathbb{X}$ has an set *complement* $\neg\mathbb{X}$ such that

$$\begin{aligned} \mathbb{X} \cup \neg\mathbb{X} &= \mathbb{U} \\ \mathbb{X} \cap \neg\mathbb{X} &= \emptyset \end{aligned}$$

- Distributive:
  1. `Union` distributes over `Intersect`
  2. `Intersect` distributes over `Union`

$$\mathbb{X} \cup (\mathbb{Y} \cap \mathbb{V}) = (\mathbb{X} \cup \mathbb{Y}) \cap (\mathbb{X} \cup \mathbb{V})$$
$$\mathbb{X} \cap (\mathbb{Y} \cup \mathbb{V}) = (\mathbb{X} \cap \mathbb{Y}) \cup (\mathbb{X} \cap \mathbb{V})$$

Pass a Quiz: Set operations

Take a quiz on page 309 to check your understanding. You can return to here from the quiz.

Counting set expressions

Counting set expressions is similar to counting Boolean expressions.

The two Boolean expressions with no variables are `False` and `True`.

The two set expressions with no set variables are $\emptyset$ and $\mathbb{U}$.

The four Boolean expressions using one variable are `False`, `True`, P and $\neg$P.

The four set expressions using one set variable are $\emptyset$, $\mathbb{U}$, $\mathbb{X}$ and $\neg\mathbb{X}$.

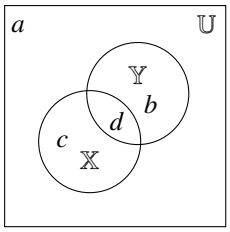
There are too many expressions on two Boolean or set variables to list here. How many expressions are there?

Counting Boolean and set expressions

Boolean variables	Boolean expressions	Set variables	Set expressions
0	$2 = 2^{2^0}$	0	$2 = 2^{2^0}$
1	$4 = 2^{2^1}$	1	$4 = 2^{2^1}$
2	$16 = 2^{2^2}$	2	$16 = 2^{2^2}$
3	$256 = 2^{2^3}$	3	$256 = 2^{2^3}$
4	$65536 = 2^{2^4}$	4	$65536 = 2^{2^4}$

Draw two intersecting subsets, call them $\mathbb{X}$ and $\mathbb{Y}$, as circles inside the universal set $\mathbb{U}$. There are 4 regions that can identified. Name the four regions in a Venn diagram: a, b, c and d. These regions can be expressed as intersections of $\mathbb{X}$ and $\mathbb{Y}$ or their complements.

Partition: $\{a, b, c, d\}$



Region	Set expression
a	$\neg\mathbb{X} \cap \neg\mathbb{Y}$
b	$\neg\mathbb{X} \cap \mathbb{Y}$
c	$\mathbb{X} \cap \neg\mathbb{Y}$
d	$\mathbb{X} \cap \mathbb{Y}$

There are $16 = 2^4 = 2^{2^2}$ ways to shade or not shade these regions.

- Choose to shade none of them; representing the empty set $\emptyset$.

- Choose to shade only one of them: There are four to do this.

- Choose to shade exactly two of them: There are six ways to do this.

- Choose to shade exactly three of them: This is equivalent to choosing one not to shade, so there are four ways to choose three from four.

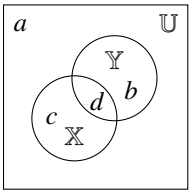
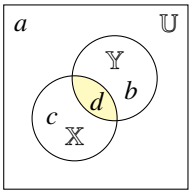
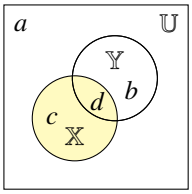
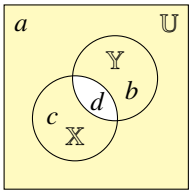
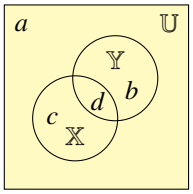
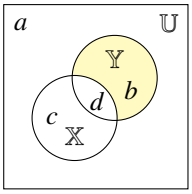
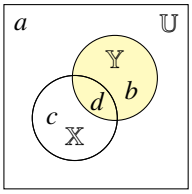
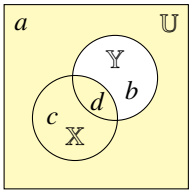
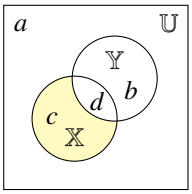
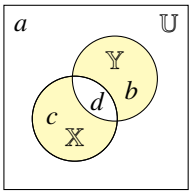
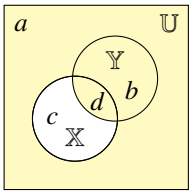
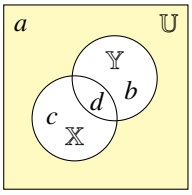
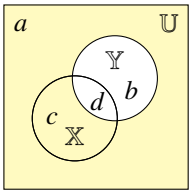
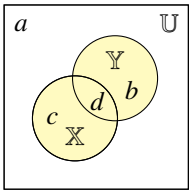
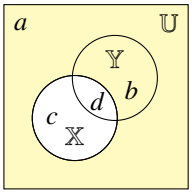
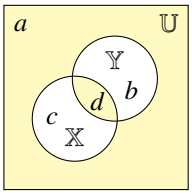
- Choose to shade all of them; representing the universal set $\mathbb{U}$.

These cases are diagrammed below. Notice the four basic regions (expressions)

$$\neg\mathbb{X} \cap \neg\mathbb{Y}, \quad \neg\mathbb{X} \cap \mathbb{Y}, \quad \mathbb{X} \cap \neg\mathbb{Y}, \quad \mathbb{X} \cap \mathbb{Y}$$

are used when shading, or not shade, the diagrams.

From the four regions $\{a, b, c, d\}$ choose

No *regions*	One *region*	Two *regions*	Three *regions*	Four *regions*
				
				
				
				
				
				

The numbers 1, 4, 6, 4, 1 are binomial coefficients from row 4 in Pascal's triangle

$$\langle 1, 4, 6, 4, 1\rangle = \left\langle \binom{4}{0}, \binom{4}{1}, \binom{4}{2}, \binom{4}{3}, \binom{4}{4} \right\rangle$$

Theorem 3: Counting Set expressions

Given n sets there are

$$2^{2^n} \quad \text{different set experssions can be formed.}$$

Proof: Counting Set expressions (sketch)

For 1 set: The two basic expressions $\mathbb{X}$ and $\neg\mathbb{X}$ partition the universal set $\mathbb{U}$.
For 2 sets: The 4 basic expressions

$$\{\neg\mathbb{X} \cap \neg\mathbb{Y},\ \neg\mathbb{X} \cap \mathbb{Y},\ \mathbb{X} \cap \neg\mathbb{Y},\ \mathbb{X} \cap \mathbb{Y}\}$$

partition the universal set $\mathbb{U}$
Introduce a new set $\mathbb{V}$. There will be 8 basic expressions that partition the universal set $\mathbb{U}$: Four when $\mathbb{V}$ intersects $\mathbb{X}$ and $\mathbb{Y}$'s basis expressions, plus four when $\neg\mathbb{V}$ does.
Create complex expressions by taking unions of the basic expressions. For the three set case, there are 8 basic expressions. To make a complex expression you can choose none of them, just 1, or 2, or 3, all the way up to all 8 of them.
Let the symbols

$$\binom{8}{0}, \binom{8}{1}, \binom{8}{2}, \binom{8}{3}, \binom{8}{4}, \binom{8}{5}, \binom{8}{6}, \binom{8}{7}, \binom{8}{8}$$

Stand for the numbers of ways to choose none, 1, 2, 3, ... 8 basic expressions from the set of 8 basic expressions over $\mathbb{X}$, $\mathbb{Y}$, and $\mathbb{V}$. The sum of these (binomial coefficients) numbers is the total number of set expressions over 3 sets.
In the notes on counting subsets you will set that

$$\binom{8}{0} + \binom{8}{1} + \binom{8}{2} + \binom{8}{3} + \binom{8}{4} + \binom{8}{5} + \binom{8}{6} + \binom{8}{7} + \binom{8}{8} = 2^8 = 2^{2^3}$$

To complete the proof, you'd note that every time a new variable is introduced, the number of basic expressions doubles. At stage n there are 2^n basic expressions. To create a complex expression, you can union any combination of the basic expressions.
To count the total number of set expressions, you need to know

$$\sum_{k=0}^{2^n} \binom{2^n}{k} = \binom{2^n}{0} + \binom{2^n}{1} + \cdots \binom{2^n}{2^n} = 2^{2^n}$$

And that link will be made in the notes on counting subsets.

Pass a Quiz: Venn diagrams

Take a quiz on page 310 to check your understanding. You can return to here from the quiz.

Homework Questions

Use your time outside of class to solve these problems.

1 Which of the following propositions are `True` and which are `False`.

1.1 $0 \in \mathbb{D}$

1.2 $(0 \in \mathbb{B}) \wedge (0 \in \mathbb{D})$

1.3 $2 \in \mathbb{B}$

1.4 $-7 \notin \mathbb{Z}$

1.5 $e \notin \mathbb{Q}$, Euler's number $e \approx 2.718\cdots$.

1.6 $\pi \in \mathbb{R}$, pi $\approx 3.1415\cdots$.

1.7 $\pi \in \mathbb{D}$

1.8 $\gamma \in \mathbb{Q}$ Euler's constant gamma $\approx 0.5772\cdot$.

2 Let $\mathbb{EVEN} = \{0, 2, 4, 6, 8\}$, $\mathbb{ODD} = \{1, 3, 5, 7, 9\}$ and $\mathbb{PRIME} = \{2, 3, 5, 7\}$ be the *even*, *odd* and *prime* digits. Compute the following set operations over the universe of digits $\mathbb{U} = \mathbb{D} = \{0, 1, 2, \ldots, 8, 9\}$.

2.1 $\neg\mathbb{EVEN}$

2.2 $\neg\mathbb{ODD}$

2.3 $\neg\mathbb{PRIME}$

2.4 $\mathbb{ODD} \cup \mathbb{PRIME}$

2.5 $\mathbb{EVEN} \cup \mathbb{PRIME}$

2.6 $\mathbb{EVEN} \cup \mathbb{ODD}$

2.7 $\mathbb{ODD} \cap \mathbb{PRIME}$

2.8 $\mathbb{EVEN} \cap \mathbb{PRIME}$

2.9 $\mathbb{EVEN} \cap \mathbb{ODD}$

2.10 $\mathbb{ODD} \cap \neg\mathbb{PRIME}$

3 How many different set expressions can be formed from three sets $\mathbb{X}$, $\mathbb{Y}$, and $\mathbb{V}$ by using union, intersection, and set complement operations?

4 How many different set expressions can be formed from four sets $\mathbb{X}$, $\mathbb{Y}$, $\mathbb{V}$, and $\mathbb{W}$ by using union, intersection, and set complement operations?

5 Answer the following `True` or `False` questions. Explain your answers.

5.1 $\emptyset = \{\emptyset\}$

5.2 $\emptyset \in \{\emptyset\}$

5.3 $\emptyset \subseteq \emptyset$

5.4 $\mathbb{X} \nsubseteq (\mathbb{X} \cup \mathbb{Y})$?

5.5 $\mathbb{X} \cap \mathbb{Y} \subseteq \mathbb{X}$?

6 Prime, odd, and triangular numbers are interesting subsets of the natural numbers $\mathbb{N}$.

- *Prime* numbers have exactly two divisors. The prime digits are

$$\mathbb{P} = \{2, 3, 5, 7\}.$$

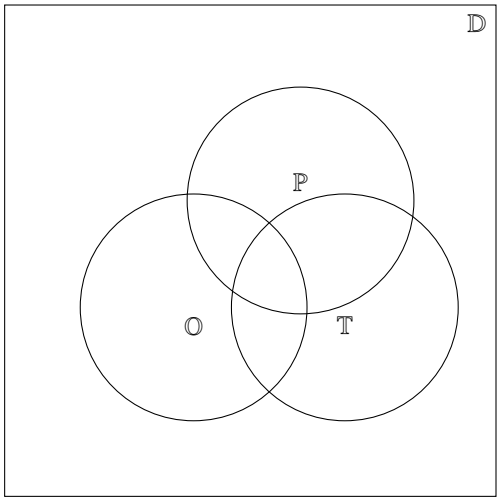
- *Odd* numbers have the form $2n + 1$ for $n \in \mathbb{N}$. The odd digits are

$$\mathbb{O} = \{1, 3, 5, 7, 9\}.$$

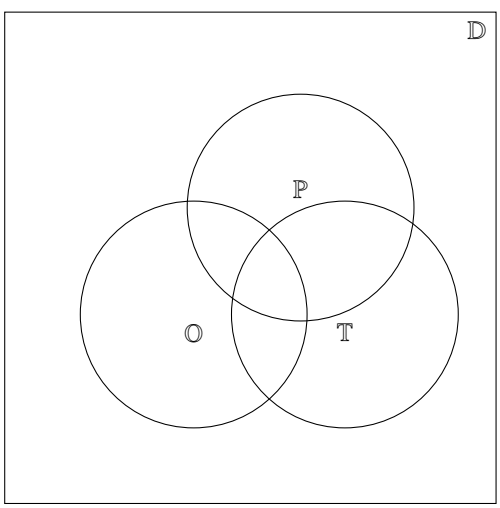
- *Triangular* numbers have the form $n(n-1)/2$ for $n \in \mathbb{N}$. The triangular digits are

$$\mathbb{T} = \{0, 1, 3, 6\}.$$

6.1 Fill in the Venn diagram below using the elements $\mathbb{P}$, $\mathbb{O}$ and $\mathbb{T}$.



7 Shade the region $(\mathbb{P} \cap \mathbb{O}) \cup \mathbb{T}$.



8 Composite, even, and Fibonacci numbers are interesting subsets of the natural numbers $\mathbb{N}$.

- *Composite* numbers are greater than 0 and have more than two divisors. The composite digits are

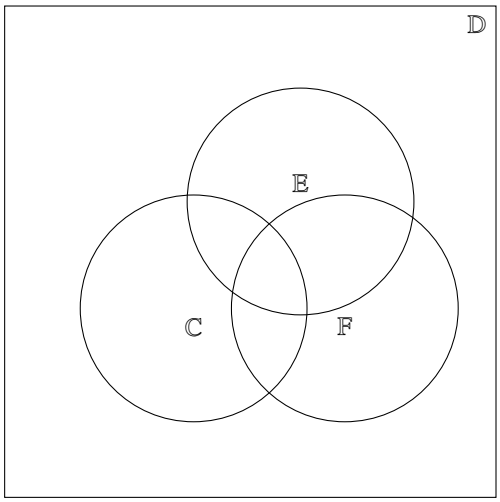
$$\mathbb{C} = \{4, 6, 8, 9\}.$$

- *Even* numbers have the form $2n$ for $n \in \mathbb{N}$. The even digits are

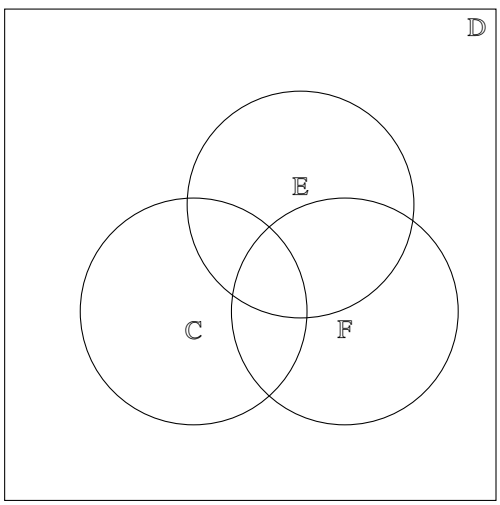
$$\mathbb{E} = \{0, 2, 4, 6, 8\}.$$

- *Fibonacci* numbers are a little more difficult to explain, but the Fibonacci digits are
$$\mathbb{F} = \{0,\ 1,\ 2,\ 3,\ 5,\ 8\}.$$

8.1 Fill in the Venn diagram below using the elements $\mathbb{C}$, $\mathbb{E}$ and $\mathbb{F}$.



8.2 Shade the region $(\mathbb{C} \cup \mathbb{E}) \cap \mathbb{F}$.



9 This problem comes from a logic puzzle attributed to Lewis Carroll.

- Let the universal set $\mathbb{U}$ be the set of all my children.
- Let $\mathbb{F}$ be my fat children and let $\mathbb{T}$ be my thin children.
- Let $\mathbb{D}$ be my daughters and let $\mathbb{S}$ be my sons.
- Let $\mathbb{H}$ be my healthy children, let $\mathbb{E}$ be my children who exercise, and let $\mathbb{G}$ be my children who are gluttons.

Using these sets and set notation ($\subseteq$, $\neg$, $\cap$, $\cup$, $\emptyset$, ...) draw diagrams to express Carroll's statements.

9.1 “All my sons are slim (thin)”.

9.2 “No child of mine is healthy who takes no exercise.”

9.3 “All gluttons, who are children of mine, are fat.”

9.4 “No daughter of mine takes any exercise.”

4. Logic: To control

Control structures

The Böhm-Jacopini theorem states only three fundamental *control* structures are necessary to implement any Turing algorithm. They are: *sequence*, *selection* and *iteration*.

Sequence

Execute instructions sequentially one after another as shown in figure 3. A sequence of instructions needs little control: Simply add 1 to the program counter and execute the instruction found at the new address.

Selection

Select one of two instructions to execute. The decision is based on the value of a Boolean variable as in figure 4.

If-then is a simple selection statement: It selects to execute an instruction or it simply responds `True`. *If-then* can be expressed using Boolean logic.

$$\textit{If } \mathrm{P}, \textit{ then } \mathrm{Q} \quad \text{is equivalent to} \quad \neg \mathrm{P} \vee \mathrm{Q}$$

- When $\mathrm{P} = \texttt{True}$, statement Q will be evaluated. In this case, the value of Q is the value of the implication *if* P, *then* Q.

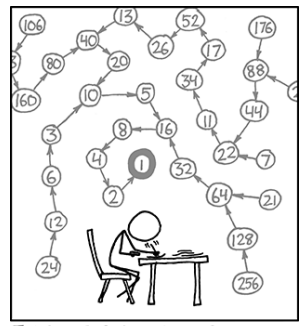
$$\textit{If } \texttt{True}, \textit{ then } \mathrm{Q} \quad \text{is equivalent to} \quad \mathrm{Q}$$

- When $\mathrm{P} = \texttt{False}$, statement Q need not be evaluated. In this case, the value of *if* P, *then* Q is simply `True`.

$$\textit{If } \texttt{False}, \textit{ then } \mathrm{Q} \quad \text{is equivalent to} \quad \texttt{True}$$

If-then-else is a complete selection statement. It describes what statement to execute when P is `True` and what alternative statement to execute when P is `False`.

$$\textit{If } \mathrm{P}, \textit{ then } \mathrm{Q}, \textit{ else } \mathrm{R} \quad \text{is equivalent to} \quad (\mathrm{P} \wedge \mathrm{Q}) \vee (\neg \mathrm{P} \wedge \mathrm{R})$$



Collatz conjecture

See the Google doodle celebrating Allen Turing's 100th birthday.

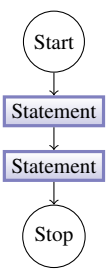


Figure 3: Sequential control

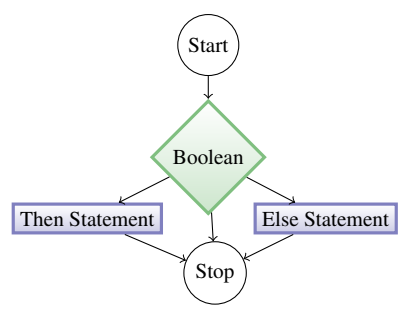


Figure 4: Selection (Conditional) control

Not evaluating Q in $\neg \mathrm{P} \vee \mathrm{Q}$ when $\mathrm{P} = \texttt{False}$ is short-circuiting the evaluation, called *lazy evaluation*.

If-then-else can be implemented using the logic shown in The truth table below

If-then-else truth table

If P, then Q, else R

	Input			Output
Row	P	Q	R	$(P \Rightarrow Q) \wedge (\neg P \Rightarrow R)$
1	0	0	0	0
2	0	0	1	1
3	0	1	0	0
4	0	1	1	1
5	1	0	0	0
6	1	0	1	0
7	1	1	0	1
8	1	1	1	1

$(\neg P \vee Q) \wedge (P \vee R)$

Predicates

Predicates are expressions that return `True` or `False` when evaluated. The returned value is predicated on the value(s) of the input variable(s). Some instances of predicates are:

- $(n = 0)$ is `True` or `False` depending on the value of n.
- $(n < m)$ is `True` or `False` depending on the values of n and m.
- $(\text{isPrime}(n))$ is `True` or `False` depending on whether or not n is a prime number.

Predicates are defined over a universal set $\mathbb{U}$. In applications, the universe of values may be very complex, but here we can assume the values over which the predicate is tested are stored in a computable list $\vec{A}$ or computable set $\mathbb{A}$.

It is useful to recognize when you do not have sufficient information to determine whether or not a predicate is `True`. For instance, the predicate (equation)

$$P(x, y) \equiv x^2 + y^2 = 1$$

remains *undecidable* until you *bind* values to x and y. Once given values, you can compute if (x, y) is on the unit circle or not. On the other hand, there are some easy to understand predicates that just cannot be decided. And some where we cannot decide if we can decide.

Iteration

Iteration loops through a sequence of instructions until some Boolean value changes as shown in figure 5. Iteration steps (loops) through predicate evaluations until one becomes `False` (or `True`). Two common programming control structures that implement iteration are **`for`** and **`while`** loops.

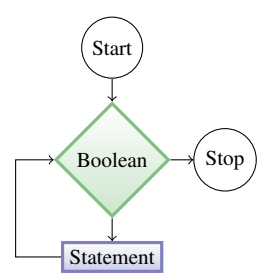


Figure 5: Iterative control

Iteration opens up the possibility of never ending computations.

Pass a Quiz: Basic predicate logic

Take a quiz on page 310 to check your understanding. You can return to here from the quiz.

Summation

A **for** loop could be used to sum the values stored in an array

$$a[k],\ k = 0,\ 1, \ldots, (n-1), \quad \vec{A} = \langle a_0,\ a_1, \ldots,\ a_{n-1} \rangle$$

A `sum()` function that totals the values in a list, when implemented in C might look like this:

```
int sum(int a[], int n) {
  int total =  0;
  for (int k = 0; k < n; k++) {
    total = total + a[k];
  }
  return total;
```

This code logic can be interpreted as the quantified expression

$$(\forall k \in \mathbb{Z}_n)((s(-1) = 0) \wedge (s(k) = s(k-1) + a[k]))$$

Basic sums you must know:

$$\sum_{k=0}^{n-1} 1 = n$$

$$\sum_{k=0}^{n-1} k = \frac{n(n-1)}{2}$$

$$\sum_{k=0}^{n-1} r^k = \frac{r^n - 1}{r - 1},\ r \neq 1$$

The worst case time complexity of both the `sum()` and `search()` functions is $O(n)$. That is, no more than cn instructions are executed, for some constant c. The `search()` function has a short circuit allowing a return once the value is found.

The set of *modular integers* $\mathbb{Z}_n$ is

$$\mathbb{Z}_n = \{0,\ 1,\ 2, \ldots,\ (n-1)\}$$

Linear search

A **for** loop could also be used to search list $\vec{a}$ for a value x.

```
bool search(int x, int a[], int n) {
  for (int k = 0; k < n; k++) {
    if (x == a[k]) {return true;}
  }
  return false;
}
```

Recursion can be used to implement `search`. For instance, in HASKELL, you could define

```
search :: (Eq x) => (x -> [x]) -> Maybe
search _ [] = \Nothing -- nothing can be found in an empty list
search x (y:ys) = if x == y then Just x else search x ys
```

which can be written in mathematical notation as

$$s(x, \mathbb{Y}) = (\exists\, y \in \mathbb{Y})(x = y)$$

Greatest common divisor

You could use a `while` loop to compute the greatest common divisor of natural numbers $n \leq m$.

```
int gcd(unsigned m, unsigned a) {  // assert 0 <= m <= a
  while (a != 0)
  {
    r = a % m;  // a = m * floor(a/m) + r = m * q + r, 0 <= r < m
    a = m;      // reduce the size of a
    m = r;      // reduce the size of m
  }
}
```

The above `gcd` algorithm terminates because the magnitude of `n` decreases with each iteration. Therefore, the terminating condition $n = 0$ will be reached.

Collatz's conjecture

A sequences

$$\vec{C} = \langle c_0, c_1, c_2, \ldots \rangle$$

is called Collatz if it is defined by the algorithm described below.

- Let $c_0 > 1$ be a natural number, called the *seed*.
- Define successive terms c_n for $n > 0$ by two cases:

$$c_n = \begin{cases} 3c_{n-1} + 1 & \text{if } c_{n-1} \text{ is is odd} \\ \frac{c_{n-1}}{2} & \text{if } c_{n-1} \text{ is even} \end{cases}$$

For instance, the Collatz sequence with seed $c_0 = 7$ is

$$\langle 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1 \rangle$$

The HASKELL code below can be used to construct a Collatz sequence.

```
Collatz :: Integer -> Integer
Collatz 1 = []
Collatz n
    | odd n = 3*n+1 : Collatz (3*n+1)
    | even n = n 'div' 2 : Collatz (n 'div' 2)
```

The Collatz conjecture states:

Conjecture 1: The Collatz conjecture

For every seed $c_0 > 1$, the Collatz sequence halts for some finite n with $c_n = 1$.

You can write this conjecture using predicate notation. It is helpful to use function $c_n = \text{collatz}(c_{n-1})$ that returns the next Collatz number. The Collatz conjecture is

$$(\forall\, c_0 > 1)(\exists\, n > 0)(\text{collatz}(c_n) = 1)$$

Quantification

PREDICATE LOGIC REASONS OVER COLLECTIONS OF THINGS. This contrasts with Boolean logic, which reasons over a single instance. Predicate logic adds two *quantifiers* to propositional logic. The quantifiers are *something* and *everything*. Predicate statements are written using the templates:

- *Some* thing satisfies a property (the predicate)
- *Every* thing satisfies a property (the predicate)

Not something is nothing. Not everything is something or nothing.

Of course, if a property is `True` for *all* values of its variable, then it is `True` for some value of its variable.

$$(\forall\, x)(\text{P}(x)) \Rightarrow (\exists\, x)(\text{P}(x))$$

For instance, you know *there is some* real number x that satisfies the equation

$$x^2 - x - 1 = 0$$

There are, in fact, two solutions. The golden ratio $\varphi = (1 + \sqrt{5})/2$ and its *conjugate* $\overline{\varphi} = (1 - \sqrt{5})/2$.

On the other hand, you know *every* real number x satisfies the equation

$$x + 0 = x$$

And, there are *no* real numbers x that satisfy the equation

$$x^2 + x + 1 = 0$$

Mathematicians use standard symbols for these and similar phrases.

$\exists$ is read: *For some*, *there exists*, *there is a least one*, or like phrases. *For some* is an existential quantifier: It can be used to state a predicate is sometimes `True` or sometimes `False`.

Some people use the symbol $\exists!$ to say "there is one and only one," that is, the thing that satisfies the predicate *unique*.

$\forall$ is read: *For all*, *for every*, *for each*, or like phrases. *For all* is a universal quantifier: It can be used to state a predicate is always `True` or never `True`.

Quantifiers modify a predicate: a statement that may be `True` or may be `False` depending of the values bound to variables in the statement.

Statements in predicate logic have forms such as

All S are T ($\mathbb{S} \subseteq \mathbb{T}$)

No S are T ($\mathbb{S} \cap \mathbb{T} = \emptyset$)

Some S are T ($\mathbb{S} \cap \mathbb{T} \neq \emptyset$)

Some S are not T ($\mathbb{S} \nsubseteq \mathbb{T}$)

Predicates arise in computing as expressions over collections of values that control the flow of a program. Predicates state relations between functions, variables, constants, things in general. For instance,

- "Point (x, y) lies on the line determined by parameters a, b, and c" can be written

  $$ax + by = c$$

  and named $\textsc{l}(x, y, a, b, c)$.

  Consider the line predicate $\textsc{l}(x, y, a, b, c)$ Fix the parameters: $a = 2$, $b = 3$ and $c = 7$ so the equation is

  $$2x + 3y = 7$$

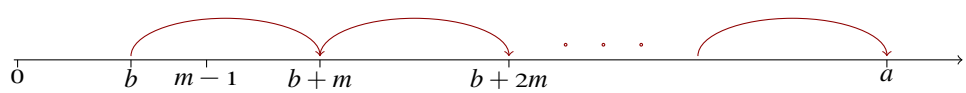
  For the point $(x, y) = (2, 1)$ the predicate is `True`, but for $(x, y) = (1, 5)$ the predicate is `False`.

- "Integers a and b are congruent mod m" can be written

  $$a - b = cm$$

  and named $\textsc{m}(a, b, m, c)$.

From b to a in jumps of m



- "$\mathbb{X}$ is a subset of $\mathbb{Y}$" can be written

  $$\mathbb{X} \subseteq \mathbb{Y}$$

  and named $\textsc{s}(\mathbb{X}, \mathbb{Y})$.

- "The value y is the output of function f given input x" can be written

  $$f(x) = y$$

  and named $\textsc{v}(f, x, y)$.

You know some functions.

- Constant functions: $y = f(x) = c$, for instance $y = 7$
- Linear functions: $y = f(x) = mx + b$, for instance $y = 3x + 2$
- Quadratic functions: $y = f(x) = ax^2 + bx + c$, for instance $y = x^2 - x - 1$
- Exponential functions: $y = f(x) = b^x$, for instance $y = 2^x$
- Logarithmic functions: $y = f(x) = \log_b x$, for instance $y = \ln x$

- "Function f maps different input x_0 and x_1 to different output values" can be written

$$(\forall\, x_0,\, x_1)((x_0 \neq x_1) \Rightarrow (f(x_0) \neq f(x_1)))$$

  and named $\textsc{OneToOne}(f)$.

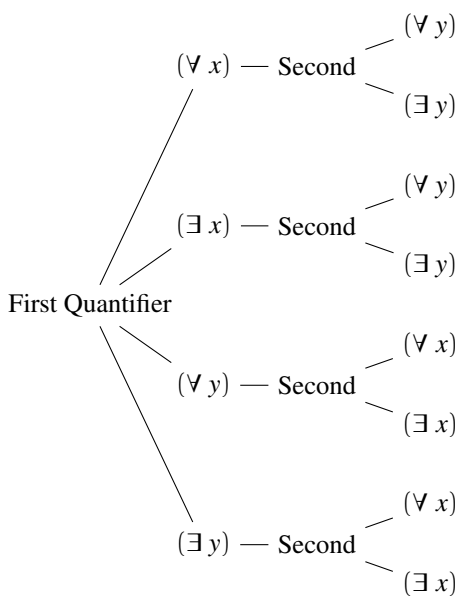
Describing the set of values where a predicate is `True` (or `False`) leads to useful definitions, theorems, and concepts.

In computing practice, the sets of things reasoned about in predicate statements belong to some type, maybe:

- athletes
- mammals
- natural numbers
- green things
- logicians
- rock bands
- functions on the integers
- permutations
- floating point numbers

Interchanging (commuting) quantifiers

One way to visualize the meaning of multiple quantifiers is through the use of graphs. It looks like there 8 combinations of $(\forall\, x)$, $(\exists\, x)$, $(\forall\, y)$, and $(\exists\, y)$, see the *decision tree* below.



However, some of these orders produce equivalent results. In fact, there are only 6 different ways to write pairs of these quantifiers.

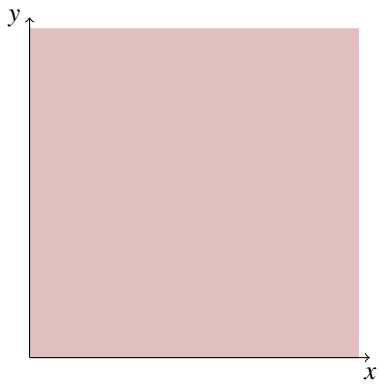
- $(\forall\, x)(\forall\, y)$ commute:

  If predicate $\textsc{p}(x,\, y)$ is `True` for every x and for every y, then it is `True` for every y and for every x.

$$(\forall\, x)(\forall\, y) \equiv (\forall\, y)(\forall\, x)$$

  The graph below shows this equivalence.

For all x's, for all y's
$(\forall\, x)(\forall\, y) \equiv (\forall\, y)(\forall\, x)$



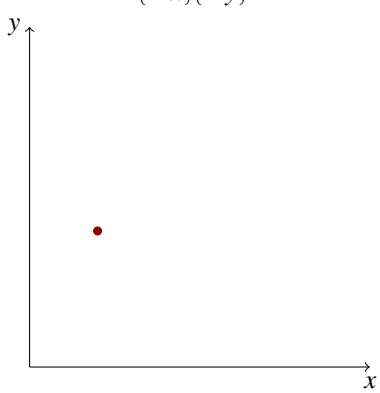
- $(\exists\, x)(\exists\, y)$ commute:

  If predicate $\textsc{p}(x,\ y)$ is `True` for some x and for some y, then it is `True` for some y and for some x.

$$(\exists\, x)(\exists\, y) \equiv (\exists\, y)(\exists\, x)$$

  The graph below gives a minimal example of this equivalence.

There exists an x, there exists a y
$(\exists\, x)(\exists\, y)$



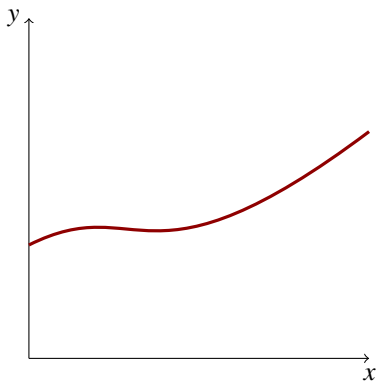
- $(\forall\, x)(\exists\, y)$ do not commute:
  If a predicate $\textsc{p}(x,\ y)$ is `True` for all x's and for some y, then it may not be `True` for some y and all x's.

In the uncommon case, when there is only one value of y (the graph cannot wiggle), then the two expressions are the same.

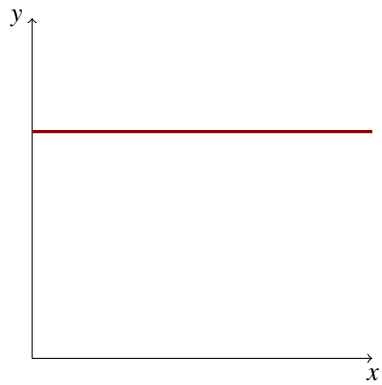
$$(\forall\, x)(\exists\, y) \not\equiv (\exists\, y)(\forall\, x)$$

  The graph below shows these orders of quantifiers are not, in general, equivalent.
  $(\forall\, x)(\exists\, y)$ allows the value of y to change depending on x.
  $(\exists\, y)(\forall\, x)$ forces y to be the same for all values of x.

For all x's, there exists a y
$(\forall\, x)(\exists\, y)$



There exists a y, for all x's
$(\exists\, y)(\forall\, x)$



- $(\forall\, y)(\exists\, x)$ do not commute:
  If a predicate $\textsc{p}(x, y)$ is `True` for all y's and for some x, then it may not be `True` for some x and all y's.

In the uncommon case, when there is only one value of x (the graph cannot wiggle), then the two expressions are the same.

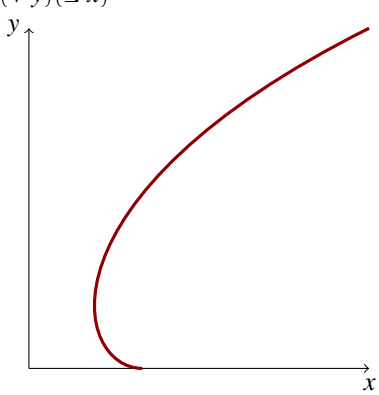
$$(\forall\, y)(\exists\, x) \not\equiv (\exists\, x)(\forall\, y)$$

The graph below shows these orders of quantifiers are not, in general, equivalent.

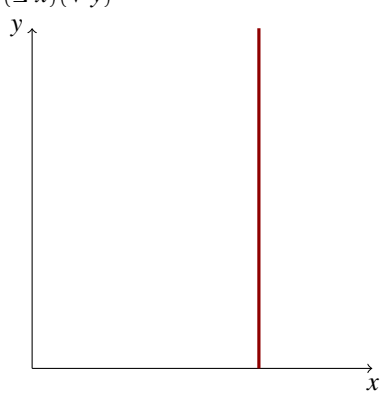
$(\forall\, y)(\exists\, x)$ allows the value of x to change depending on y.

$(\exists\, x)(\forall\, y)$ forces x to be the same for all values of y.

For every y, there is an x
$(\forall\, y)(\exists\, x)$



There is an x for every y
$(\exists\, x)(\forall\, y)$



Pass a Quiz: Predicates: Logic for control

Take a quiz on page 311 to check your understanding. You can return to here from the quiz.

Homework Questions

🏠 Use your time outside of class to solve these problems.

These "distribute not across ..." exercises are De Morgan-like laws for first-order (predicate) logic.

1 Rewrite the following expressions by distributing NOT over the quantifier.

1.1 $\neg(\forall\, x)(\mathrm{P}(x))$,

1.2 $\neg(\exists\, x)(\mathrm{P}(x))$.

1.3 $\neg(\forall\, x)(\exists\, y)(\mathrm{P}(x,\, y))$.

1.4 $\neg(\exists\, x)(\exists\, y)(\mathrm{P}(x,\, y))$.

1.5 $\neg(\forall\, x)(\forall\, y)(\mathrm{P}(x,\, y))$.

Commutative-like laws are also basic: Some systems obey them, others don't. Those systems that have commutative laws are called Abelian. It is useful to know how $\forall$ and $\exists$ commute or don't commute.

2 How do $\forall$ and $\exists$ commute? That is, which of the equivalences below are `True` and which are `False`?

2.1 $(\forall\, x)(\forall\, y)(\mathrm{P}(x,y)) \equiv (\forall\, y)(\forall\, x)(\mathrm{P}(x,y))$

2.2 $(\exists\, x)(\forall y)(\mathrm{P}(x,y)) \equiv (\forall\, y)(\exists\, x)(\mathrm{P}(x,y))$

2.3 $(\exists\, x)(\exists\, y)(\mathrm{P}(x,y)) \equiv (\exists\, y)(\exists\, x)(\mathrm{P}(x,y))$

2.4 $(\forall\, x)(\exists\, y)(\mathrm{P}(x,y)) \equiv (\exists\, y)(\forall\, x)(\mathrm{P}(x,y))$

3 Explain what the statements below says about the function $f : \mathbb{X} \to \mathbb{Y}$.

3.1 $(\forall\, x \in \mathbb{X})(\forall\, y_0,\, y_1 \in \mathbb{Y})((f(x) = y_0) \wedge (f(x) = y_1) \Rightarrow (y_0 = y_1))$

3.2 $(\forall\, y \in \mathbb{Y})(\exists\, x \in \mathbb{X})(f(x) = y)$

3.3 $(\forall\, x_0,\, x_1 \in \mathbb{X})((x_0 \neq x_1) \Rightarrow (f(x_0) \neq f(x_1)))$

3.4 $(|\mathbb{X}| > |\mathbb{Y}|) \Rightarrow (\exists\, x_0,\, x_1 \in \mathbb{X})((x_0 \neq x_1) \wedge (f(x_0) = f(x_1)))$

I found the $\textsc{canfool}(p,\, t)$ problem in Christos Papadimitriou's book on computational complexity (Papadimitriou, 1994)

4 Let $\textsc{canfool}(p,\, t)$ be the proposition

> "*You can fool person p at time t.*"

Use quantifiers ($\forall$, $\exists$) over the set of persons $\mathbb{P}$ and the set of time $\mathbb{T}$ to write the following sentences using the $\textsc{canfool}(p,\, t)$ predicate.

4.1 *You can fool some of the people all of the time.*

4.2 *You can fool all of the people some of the time.*

4.3 *You cannot fool all of the people all the time.*

4.4 Name your answers to parts 4.1, 4.2 and 4.3 P_a, P_b, and P_c to write the famous quotation attributed to Abraham Lincoln.

> *You can fool all of the people some of the time, and some of the people all the time, but you cannot fool all of the people all the time.*

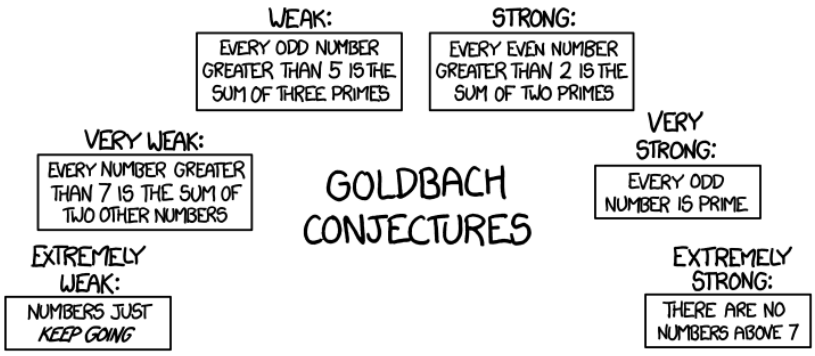
5 Let $\mathbb{F} = \{f : \mathbb{N} \to \mathbb{N}\}$ be the set of functions mapping the natural numbers to the natural numbers. Given a function $f \in \mathbb{F}$, big O of f, written $O(f)$, is the set of functions $g \in \mathbb{F}$ such that $g(n)$ is less than or equal to $c \cdot f(n)$ for some constant $c > 0$ and for all n greater than some natural number N.

Using the notation of predicate logic, write a statement that describes when $g \in O(f)$

6 Fermat's last theorem states: If $n \geq 3$, then there are no positive integer x, y and z that solve the equation

$$\boxed{x^n + y^n = z^n}$$

Write Fermat's last theorem using the language of predicates and quantifiers.



Goldbach Conjectures

7 The Goldbach's conjecture comes from noticing the pattern:

$4 = 2 + 2$	$6 = 3 + 3$	$8 = 3 + 5$
$10 = 5 + 5$	$12 = 5 + 7$	$14 = 7 + 7$
$16 = 5 + 11$	$18 = 5 + 13$	$20 = 7 + 13$

It appears that every even integer greater than 2 can be written as the sum of two prime numbers.

Conjecture 2: Goldbach's conjecture

Every even integer greater than 2 can be written as the sum of two prime numbers.

Write Goldbach's conjecture using the notation of predicate logic.

8 The Twin prime conjecture comes from the pattern

$(3, 5)$	$(5, 7)$	$(11, 13)$
$(17, 19)$	$(29, 31)$	$(41, 43)$
$(59, 61)$	$(71, 73)$	$(101, 103)$

It appears there is a never-ending list of "twin primes:" Pairs (p, q) where both p and q are prime numbers and $q = p + 2$.

Conjecture 3: Twin prime conjecture

There are infinitely many prime numbers p such that $p + 2$ is also prime.

Write the Twin prime using the notation of predicate logic.

9 What is the Collatz sequence for the seed $c_0 = 2$?

10 What is the Collatz sequence for the seed $c_0 = 13$?

11 Use mathematical notation to write the predicate "The weak can never forgive," that is attributed to Mahatma Ghandi.

12 The statements below come from Lewis Carroll (Carroll, 1958). For each statement, write a quantified predicate statement.

12.1 No Frenchmen like plum pudding.

12.2 All Englishmen like plum pudding.

12.3 Some thin persons are not cheerful.

12.4 All pigs are fat.

12.5 Some lessons are difficult.

12.6 All clever people are popular.

12.7 Some healthy people are fat.

13 Let $\text{P}(x)$ be the statement $x^2 = x + 1$. What is the truth value of the following statements.

13.1 $\text{P}(0)$

13.2 $\text{P}((1 + \sqrt{5})/2)$

13.3 $(\forall x)(\text{P}(x))$

13.4 $(\forall x)(\neg\text{P}(x))$

13.5 $(\exists x)(\text{P}(x))$

13.6 $(\exists x)(\neg\text{P}(x))$

14 Write the Abel-Ruffini theorem using the notation of predicate logic.

Theorem 4: Abel-Ruffini

There is no single formula for the roots of a fifth (or higher) degree polynomial in terms of its coefficients, using only the usual algebraic operations (addition, subtraction, multiplication, division) and application of radicals (square roots, cube roots, etc).

There is the quadratic formula for the roots of $ax^2 + bx + c = 0$, And, complicated formulas have been found for cubic ($ax^3 + bx^2 + cx + d = 0$) and quartic ($ax^4 + bx^3 + cx^2 + dx + e = 0$) equations. In the search for a formula for solutions of fifth degree polynomial equations, people eventually found there was none.

15 Formal methods rely on a few basic concepts.

- A *pre-condition* is a predicate known to be `True` before a function (program, code segment, machine, network, exchange, ...) executes.
- A *post-condition* is a predicate that can be proven to be `True` after completion of a function (program, ...), provided the pre-condition was `True`.
- An *invariant* is a condition that remains `True` before and after an algorithm structure (a loop, a recursive call, a subroutine). Invariants are used to prove post-conditions given the pre-conditions.

What are the pre-and-post-conditions for the following functions?

15.1 Sort

15.2 Linear search

15.3 Binary search

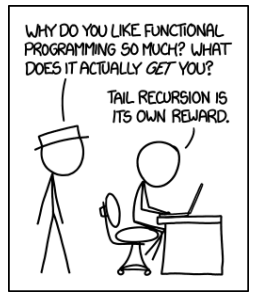
If you were to construct these functions, what invariants would you use?

Pass a Quiz: Summative exam #1 on numbers and Boolean logic

Take a quiz on page 312 to check your understanding. You can return to here from the quiz.

5. *Programming*: *With functions*

Recursive functions are one model, among many, of computation: A computer or a computer program can be thought of as a function that maps input data x of type $\mathbb{X}$ to output information y of type $\mathbb{Y}$. The programmed function must, almost always, be broken down into more simple functions that can be composed to implement the overall function (program).



Functional

A model of computation is a "way" or "method" used to represent computation. Some examples are Turing machines, recursive functions, λ calculus, and Post production systems.

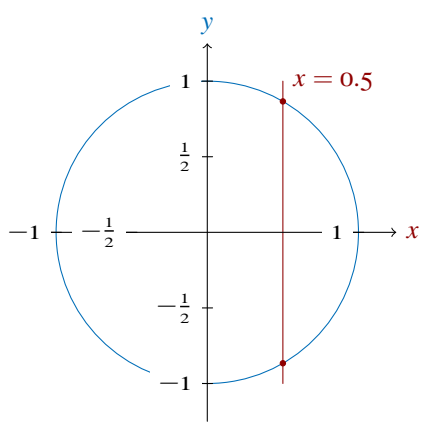
A function $f : \mathbb{X} \to \mathbb{Y}$ is *total* and *deterministic*: Given any $x \in \mathbb{X}$, there is one and only one $y \in \mathbb{Y}$ such that $f(x) = y$. *Total* means every input $x \in \mathbb{X}$ is mapped to some output $y \in \mathbb{Y}$. *Deterministic* means each input $x \in \mathbb{X}$ maps to only one output $y \in \mathbb{Y}$. The input space $\mathbb{X}$ is called the *domain*, and the output space $\mathbb{Y}$, called the *co-domain*. The set $f(\mathbb{X})$ is called the *range*.

Functions

Functions are studied in continuous mathematics: Usually by way of graphs of curves in a two-dimensional (x, y) Cartesian coordinate system. A curve represents a function if it passes the *vertical line test*:

Every vertical line crosses the graph of the curve at most once.

A full-circle $y = \pm\sqrt{1 - x^2}$ is *not* a function, but the top semi-circle $y = \sqrt{1 - x^2}$ is a function.



The vertical line $x = 0.5$ crosses the circle twice, at $y = \pm\sqrt{0.75} = \sin \pi/3 \approx 0.866025403784$. The relation $y = \pm\sqrt{1 - x^2}$ between x and y is not a function. But each separate equation

$$y = +\sqrt{1 - x^2} \quad \text{and} \quad y = -\sqrt{1 - x^2}$$

define functions.

Of course, this definition does not tell you how to *compute* f. The definition only describes a property of the map from $\mathbb{X}$ to $\mathbb{Y}$.

Definition 8: Function

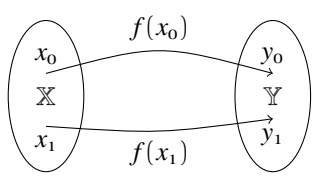
Using predicate logic, you can write $f : \mathbb{X} \to \mathbb{Y}$ is a function by

$$(\forall\, x \in \mathbb{X})(\forall\, y_0, y_1 \in \mathbb{Y})((f(x) = y_0) \wedge (f(x) = y_1) \Rightarrow (y_0 = y_1))$$

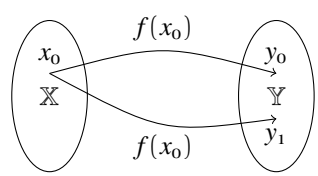
This statement says, if f maps x to two values: y_0 and y_1, then these values are, in fact, equal. Conversely, you could say, if the y's are different, the x's must be different too. By De Morgan's laws and contraposition

$$((f(x) = y_0) \wedge (f(x) = y_1)) \Rightarrow (y_0 = y_1) \equiv$$
$$(y_0 \neq y_1) \Rightarrow ((f(x) \neq y_0) \vee (f(x) \neq y_1))$$

In the diagram below: The figure on the left shows a function mapping each x to one and only one y. The figure on the right shows a relation that is not a function; some x_0 maps to different values of y.



f is a function from $\mathbb{X}$ into $\mathbb{Y}$



f is *not* a function from $\mathbb{X}$ into $\mathbb{Y}$

Properties of functions

There are several important properties of functions. Two of them are: Onto and one-to-one.

$f : \mathbb{X} \to \mathbb{Y}$ is onto when $f(\mathbb{X}) = \mathbb{Y}$. Otherwise, f only map $\mathbb{X}$ *into* $\mathbb{Y}$.

Onto property of functions

Definition 9: Onto

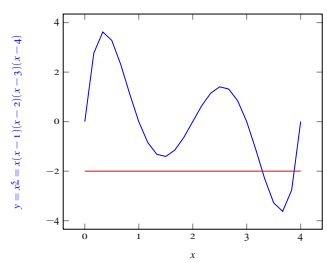
A function $f : \mathbb{X} \to \mathbb{Y}$ maps $\mathbb{X}$ onto $\mathbb{Y}$ if every $y \in \mathbb{Y}$ has an $x \in \mathbb{X}$ such that $f(x) = y$.

$$(\forall\, y \in \mathbb{Y})(\exists\, x \in \mathbb{X})(f(x) = y)$$

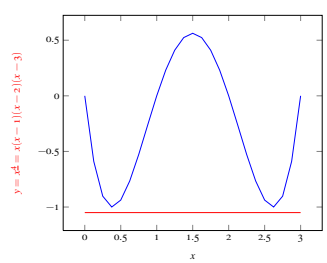
A curve is onto if it passes the *first horizontal line test*:

Every horizontal line crosses the graph of the curve at least once.

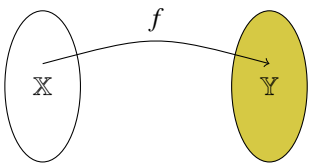
Another way to understand the onto concept is visualize a function as a map from a source set $\mathbb{X}$ to a target set $\mathbb{Y}$. An onto function completely fills the target set. In the diagram below: The figure on the left shows an onto function, each y has at least one arrow from some x mapping to it. The figure on the right shows a function that is not onto; some y's have no arrow that map to them.



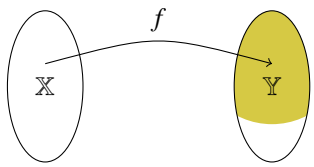
$x^5 = x(x-1)(x-2)(x-3)(x-4)$ is onto.



$x^4 = x(x-1)(x-2)(x-3)$ is *not onto* $\mathbb{R}$.



f maps $\mathbb{X}$ *onto* $\mathbb{Y}$



f does *not* map $\mathbb{X}$ *onto* $\mathbb{Y}$

In C, onto can be tested using nested loops. Consider the pseudo-code:

```
for (i=0; i< |Y|; i++) // loop over all of the y's
{                      // okay! |Y| is not legal C.
  y = Y[i];          // y is value in Y at address i

  ontoY = false;      // assume a false result
  for (j=0; j<|X|; j++) // loop over all the x's
  {
    x = X[j];
    if (f(x) == y) {     // $f$ an found y that maps to y
      ontoY = true;         // f was onto this value of y
      break;                // cut the j loop short
    }
  }
  if (ontoY == false) return false; // no x was not found for this y
}
return true;
```

This pseudo-code has worst case time complexity $O(|\mathbb{Y}| \cdot |\mathbb{X}|)$.

The inner `for` loop on `j` can execute up to $|\mathbb{X}|$ times, at a constant cost c per iteration giving a total cost of $c\,|\mathbb{X}|$. (This is an application of summing terms in the Alice sequence.)

The outer `for` loop on `i` executes $|\mathbb{Y}|$ times. The time to it takes complete one iteration is some constant cost d plus the cost of the inner loop.

In HASKELL, the function `onto` to test whether or not a function f from $\mathbb{X}$ to $\mathbb{Y}$ is onto.

```
onto :: (Eq y) => (x -> y) -> [x] -> [y] -> Bool
onto f xs [] = True -- every function maps onto the empty set
onto f xs (y:ys) = elem y (image f xs) && onto f xs ys
```

For this code two helper functions are needed.

1. `elem` that tests if a value is an element of a list

```
elem :: (Eq x) => (x -> [x]) -> Bool
elem _ [] = False -- no element is in the empty list
elem x (y:ys) = x == y || elem x ys  -- x is in the list (y:ys)
                                     -- if x is y or
                                     -- if x is in the tail ys
```

2. `image` that creates the list $f(x)$ such that $x \in \mathbb{X}$.

```
image :: => (x -> y) -> [x] -> [y]
image f xs = [f x | x <- xs]  -- we should remove duplicates
                              -- from the returned list
```

One-to-one property of functions

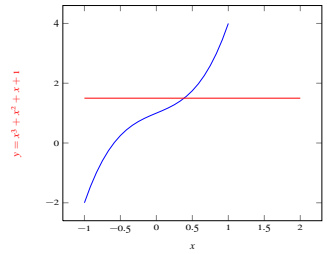
Definition 10: One-to-one

A function $f : \mathbb{X} \to \mathbb{Y}$ maps $\mathbb{X}$ one-to-one to $\mathbb{Y}$ if no $y \in \mathbb{Y}$ is the image of two or more x's in $\mathbb{X}$. That is, different x's map to different y's.

$$(\forall\, x_0, x_1 \in \mathbb{X})((x_0 \neq x_1) \Rightarrow (f(x_0) \neq f(x_1)))$$

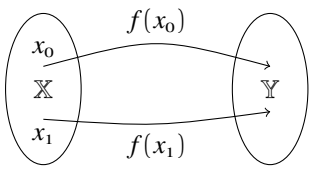
A curve is one-to-one if it passes the *second horizontal line test*:

Every horizontal line crosses the graph of the curve at most once.

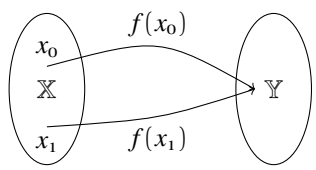


The geometric sum $y = x^3 + x^2 + x + 1$ is a one-to-one function.

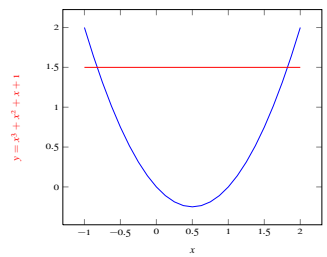
A one-to-one function never maps two (or more) source values to the same target. In the diagram below: The figure on the left shows an one-to-one function, no y has at more than one arrow mapping to it. The figure on the right shows a function that is not one-to-one; some y have more than one arrow mapping to it.



f maps $\mathbb{X}$ *one-to-one* $\mathbb{Y}$



f does *not* map $\mathbb{X}$ *one-to-one* $\mathbb{Y}$



The falling factorial function $y = x^{\underline{2}} = x(x-1)$ is not one-to-one.

In C, *one-to-one* can be tested using nested loops.

```
for (i=0; i<|X|; i++) { // loop over all of the x's
  xi = X[i];            // value in X at index i

  for (j=i+1; j<|X|; j++) { // loop over remaining x's
    xj = X[j];            // value in X at index j
    if (f(xi) == f(xj)) { // if xi and xj map to the same value
      return false;       // return f is not one-to-one
    }
  }
}
return true;
```

In Haskell, the function `oneToOne` tests whether or not a function f from $\mathbb{X}$ to $\mathbb{Y}$ is oneToOne.

```
oneToOne :: Eq b => (a -> b) -> [a] -> Bool
oneToOne f xs [] = True -- every function maps xs
                        -- one-to-one to the empty set
oneToOne f (x:xs) = not elem  (f x) (image f xs) && oneToOne f xs
```

Pass a Quiz: Basic function concepts

Take a quiz on page 314 to check your understanding. You can return to here from the quiz.

Cardinality of a set

Intuitively, the *cardinality* of a set is the number of elements in it. The symbol $|\mathbb{X}|$ is used to denote the cardinality of a set. For finite sets, this is straightforward:

Just as the absolute value symbol $|x|$ is used to denote the "size" or "magnitude" of a number, $|\mathbb{X}|$ is used to denote the size of a set.

- The cardinality of the bits $\mathbb{B} = \{0, 1\}$ is $|\mathbb{B}| = 2$.
- The cardinality of the digits $\mathbb{D} = \{0, 1, 2, \ldots, 9\}$ is $|\mathbb{D}| = 10$.
- The cardinality of the hexadecimal numberals $\mathbb{H} = \{0, 1, 2, \ldots, 9, A, B, \ldots, F\}$ is $|\mathbb{H}| = 16$.

Definition 11: Cardinality of finite sets

A finite set $\mathbb{X}$ has cardinality m if there is a one-to-one function from the modular integers $\mathbb{Z}_m = \{0, 1, 2, \ldots, (m-1)\}$ onto $\mathbb{X}$.

The cardinality of an infinite set is more difficult to understand. There are unbounded orders of infinity. The smallest infinity is the size of the natural numbers $\mathbb{N}$.

Definition 12: Cardinality of the natural numbers

The cardinality of $\mathbb{N}$ is defined to be $\aleph_0$, pronounced "aleph-naught."

Countable sets are finite or they have the same size (cardinality) as the natural numbers.

Definition 13: Cardinality of a countable set

A set $\mathbb{X}$ is countable if it is finite or there is a one-to-one function from the natural numbers $\mathbb{N}$ onto $\mathbb{X}$. In this case, the cardinality of $\mathbb{X}$ is $|\mathbb{X}| = \aleph_0$, "$aleph - naught$."

Most people find it hard to imagine that $\mathbb{N} = \{0, 1, 2, 3, 4, 5, \ldots\}$ and $\mathbb{Z} = \{0, \pm 1, \pm 2, \pm 3, \pm 4, \pm 5, \ldots\}$ contain the same number of elements: $\mathbb{Z}$ looks twice as large!

The *mediant* of rationals numbers $\frac{r}{s}$ and $\frac{u}{v}$ is $\frac{r+u}{s+v}$. An entry $\frac{i}{j}$ in the Stern-Brocot tree is the mediant of the first left ancestor and first right ancestor on the path from $\frac{i}{j}$ to the root.

I find it remarkable that the rational numbers $\mathbb{Q}$ are countable.

Example: The integers $\mathbb{Z}$ are countable

Here's how to construct a one-to-one function from $\mathbb{N}$ onto $\mathbb{Z}$. Let

$$f(n) = \begin{cases} \frac{n}{2} & \text{if } n \text{ is even} \\ -\left(\frac{n+1}{2}\right) & \text{if } n \text{ is odd} \end{cases}$$

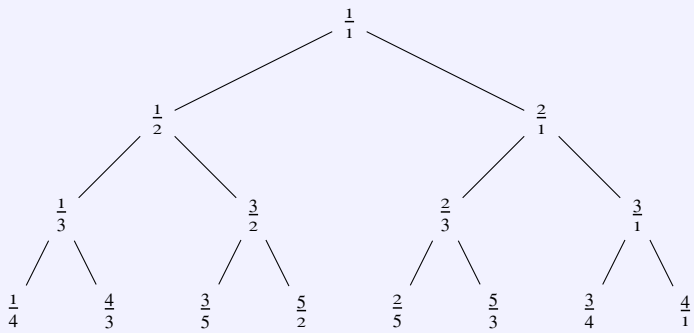
The first few mappings by this function are given in the list:

n	0	1	2	3	4	5	6	7	8	9	$\cdots$
$z = f(n)$	0	-1	1	-2	2	-3	3	-4	4	-5	$\cdots$

Every integer z has a natural number n that maps to it. If z is greater than or equal to 0, then $n = 2z$ maps to z. If z is less than 0, then $n = 2|z| - 1$ maps to z.

Example: The rationals $\mathbb{Q}$ are countable

Consider the Stern-Brocot tree: It is a binary tree *where each node is labeled $\frac{r}{s}$. The node $\frac{r}{s}$ has two children: Its left child is $\frac{r}{r+s}$ and its right child is $\frac{r+s}{s}$.*



The following statements about the entries in the Stern-Brocot tree are `True`*. See (Calkin and Wilf, 2000) for a more rigorous proofs. All node are in reduced form, that is, the numerator and denominator are* relatively prime*; their greatest common divisor is* 1*.*

Proof: Stern-Brocot: Reduced fractions

The root vertex $\frac{1}{1}$ satisfies the statement: It is reduced. Similarly, observe at the next two levels

$$\frac{1}{2} \quad \text{and} \quad \frac{2}{1} \quad \text{are reduced.}$$

and

$$\frac{1}{3}, \frac{2}{3}, \frac{3}{2}, \frac{3}{1} \quad \text{are reduced.}$$

Suppose, by way of contradiction, there is some node r/s where the fraction is not reduced. That is, some natural number larger that 1 *evenly divides both. You can assume this node is at the highest possible level.*
If r/s is a left child, then its parent is $r/(s-r)$

$$\frac{r}{s-r} \xrightarrow{\text{left}} \frac{r}{r+(s-r)} = \frac{r}{s}$$

If r/s is a right child, then its parent is $(r-s)/s$

$$\frac{r-s}{s} \xrightarrow{\text{right}} \frac{(r-s)+s}{s} = \frac{r}{s}$$

In both cases, the parent is not reduced if r/s is not reduced. And, the parent is at a higher level than r/s, contradicting the assumption that r/s was at the highest level.
Every positive rational number appears somewhere in the Stern-Brocot tree.

The rational number 1 is at the root of the tree. Suppose, by way of contradiction, there is some fraction r/s that is not in the tree.
You may assume that r/s is the one with smallest numerator and denominator. If $r > s$, then $(r - s)/s$ would not be in the tree either, but its numerator $(r - s)$ is smaller than r, a contradiction.
If $r < s$, then $r/(s - r)$ would not be in the tree either, but its denominator $(s - r)$ is smaller than s, another contradiction.
Finally, you must show no rational number appears twice in this tree. Can you suggest a proof of this?

Pigeonhole principle

Suppose you owned n pigeons and had a coop with m pigeonholes. The pigeonhole principle states: If you've more pigeons than pigeonholes you need a bigger coop.

Here there are $n = 10$ pigeons in $m = 9$ pigeonholes. The first pigeonhole has two birds.

Theorem 5: Pigeonhole Principle

If n pigeons are placed in m pigeonholes where $n > m$, then some pigeonhole will have more than one pigeon.

Expressing the pigeonhole principle as its *contrapositive* can also be useful.

If no pigeonhole contains more than one pigeon, then there are fewer pigeons than pigeonholes, that is, $n \leq m$

Here are some examples of the pigeonhole principle.

Example: Fruit, words, birthdays, and printable characters

Pretend you have three types of fruit: Bunches of apples, oranges, and grapes that you want to distribute to a group of people.

- *If there are 4 or more people, some fruit will be given out twice or more*

  $$(P_1, A), (P_2, O), (P_3, G), (P_4, ?)$$

  The first 3 people can be given different types of fruit, but the fourth person must get a fruit already given.

  Here $n = 4$ people are distributed among $m = 3$ types of fruit. Notice that $n/m = 4/3$ rounds up to 2: $\lceil 4/3 \rceil = 2$.

- *If there are 7 or more people, at least 3 people will given the same fruit.*

  $$(P_1, A), (P_2, O), (P_3, G), (P_4, A), (P_5, O), (P_6, G), (P_7, ?)$$

Notice that $7/3$ rounds up to 3: $\lceil 7/3 \rceil = 3$.

- *If there are 10 or more people, at least 4 will get the same fruit. Notice that $10/3$ rounds up to 4: $\lceil 10/3 \rceil = 4$.*
- *In general, if there are n or more people, at least $\lceil n/3 \rceil$ will get the same fruit.*
- *On the other hand, if there are 2 or fewer people, some fruit type will not be given away.*

$$(P_1, A), (P_2, O)$$

No grapes were given.

Notice that $2/3$ rounds down to 0: $\lfloor 2/3 \rfloor = 0$.

- *If there are 5 or fewer people, some fruit type will not be given away more than once.*

$$(P_1, A), (P_2, O), (P_3, G), (P_4, A), (P_5, O)$$

Grapes were given only once.

Notice that $5/3$ rounds down to 1: $\lfloor 5/3 \rfloor = 1$.

- *If there are 8 or fewer people, some fruit will not be given away more than twice.*

  Notice that $8/3$ rounds down to 2: $\lfloor 8/3 \rfloor = 2$.
- *In general, if there are n or fewer people, some fruit will be not be given away more than $\lfloor n/3 \rfloor$ times.*

Example: English words

Pretend you've collected English words.

- *In a set of 27 words, at least 2 will start with the same letter.*
- *In a set of 53 words, at least 3 will start with the same letter.*

Pretend you've collected people.

- *In a group of 367 people, at least 2 will have the same birthday.*
- *In a group of 733 people, at least 3 will have the same birthday.*

Finally, there are 96 printable characters in the ASCII character set. In any string of 97 or more printable ASCII characters, some character must appear more than once.

The pigeonhole principle can be stated in terms of onto and one-to-one functions.

- If $|\mathbb{X}| < |\mathbb{Y}|$, then there does not exist any *onto* function $f : \mathbb{X} \to \mathbb{Y}$.

  A small set cannot functionally cover a larger set. If there are fewer pigeons than holes, some hole will be empty.

  $$(|\mathbb{X}| < |\mathbb{Y}|) \Rightarrow (\forall\, f : \mathbb{X} \to \mathbb{Y})(\exists\, y \in \mathbb{Y})(\forall\, x \in \mathbb{X})(f(x) \neq y)$$

  The conclusion (right-hand side) of the above implication can be stated as: For every function mapping $\mathbb{X}$ to $\mathbb{Y}$ there is a $y \in \mathbb{X}$ and no value of $x \in \mathbb{X}$ maps to it.

- If $|\mathbb{X}| > |\mathbb{Y}|$, then there does not exist any one-to-one function $f : \mathbb{X} \to \mathbb{Y}$.

  A large set must over-cover some element in a smaller set. If there are more pigeons than holes, some hole will hold two or more pigeons.

  $$(|\mathbb{X}| > |\mathbb{Y}|) \Rightarrow (\forall\, f : \mathbb{X} \to \mathbb{Y})(\exists\, x_0,\, x_1 \in \mathbb{X},\, x_0 \neq x_1)(f(x_0) = f(x_1))$$

  The conclusion (right-hand side) of the above implication can be stated as: For every function mapping $\mathbb{X}$ to $\mathbb{Y}$ there are two different values in $\mathbb{X}$ that map to the same value in $\mathbb{Y}$.

The sum

$$\sum_{k=1}^{m} p(k) = n$$

Why is this `True`?

Theorem 6: Generalize Pigeonhole principle

Let n be the number of pigeons to be placed in m pigeonholes. Let $p(k)$ be the number of pigeons place in pigeonhole k for $k = 1, 2, \ldots, m$.

1. Some pigeonhole contains at least $\lceil n/m \rceil$ pigeons.

$$(\exists\, k \in \mathbb{Z}_m)\left(p(k) \geq \left\lceil \frac{n}{m} \right\rceil\right)$$

2. Some pigeonhole contains no more than $\lfloor n/m \rfloor$ pigeons.

$$(\exists\, k \in \mathbb{Z}_m)\left(p(k) \leq \left\lfloor \frac{n}{m} \right\rfloor\right)$$

The proof is by contraction. See the chapter on proofs by contradiction for additional examples of this technique.

Proof: Generalized Pigeonhole principle

Let $p(k)$ be the number of pigeons in hole k for $k = 0, 1, 2, \ldots, (m-1)$. Summing $p(k)$ over all holes gives a total of n.

$$n = \sum_{holes} |\textit{pigeons-in-hole}| = \sum_{k=0}^{m-1} p(k)$$

1. *If every pigeonhole contains fewer than $\lceil n/m \rceil$ pigeons, then, $p(k) \leq \lceil n/m \rceil - 1$ for all $k = 0, 1, \ldots, (m-1)$. Summing $p(k)$ over each hole gives,*

$$n = \sum_{k=0}^{m-1} p(k) \leq \sum_{k=0}^{m-1} \left(\left\lceil \frac{n}{m} \right\rceil - 1\right) = m\left\lceil \frac{n}{m} \right\rceil - m$$

Notice that

$$\left\lceil \frac{n}{m} \right\rceil < \frac{n}{m} + 1$$

Therefore,

$$n \leq m \left\lceil \frac{n}{m} \right\rceil - m < m\left(\frac{n}{m} + 1\right) - m = n$$

A contradiction.

2. *If every pigeonhole contains more than $\lfloor n/m \rfloor$ pigeons, then, $p(k) \geq \lfloor n/m \rfloor + 1$ for all $k = 0, 1, \ldots, (m-1)$. Summing $p(k)$ over each hole gives,*

$$n = \sum_{holes} |pigeons\text{-}in\text{-}hole| \geq \sum_{holes} \left(\left\lfloor \frac{n}{m} \right\rfloor + 1\right) = m \left\lfloor \frac{n}{m} \right\rfloor + m$$

Notice that

$$\left\lfloor \frac{n}{m} \right\rfloor > \frac{n}{m} - 1$$

Therefore,

$$n \geq m \left\lfloor \frac{n}{m} \right\rfloor + m > m\left(\frac{n}{m} - 1\right) + m) = n$$

A contradiction.

Here are some additional pigeonhole principle examples.

Example: Hash Tables

A hash table is a data structure used in many computing applications. Intuitively, a hash table is a collection of boxes where objects are stored. If there are more objects than boxes, then some box must contain more than one object.
If there are m boxes in the hash table and n objects to store, then

- *Some box will contain $\lceil n/m \rceil$ or more objects.*

  Let there be $m = 5$ boxes in which to put $n = 12$ objects. Then some box will contain $\lceil 12/5 \rceil = 3$ objects. (If each box contained 2 or fewer objects there would be no more than 10 objects.)

- *Some box will contain $\lfloor n/m \rfloor$ or fewer objects.*

  For the $m = 5$ box, $n = 12$ object example, some box contains $\lfloor 12/5 \rfloor = 2$ or fewer objects. (If each box contained 3 or more objects there would be more than 15 objects.)

Pass a Quiz: Pigeonhole principle

Take a quiz on page 314 to check your understanding. You can return to here from the quiz.

Classification of functions

THERE ARE SEVERAL USEFUL FUNCTION CLASSES: Polynomials, logarithms and exponentials, permutations, and integer functions are some of these.

Polynomials

A polynomial can be written as a weighted sum of powers of x. For instance,

$$\begin{aligned} p(x) &= x^2 - x - 1 && \text{weights (coefficients) } \langle -1,\ -1,\ 1\rangle \\ q(x) &= x^3 + x^2 + x + 1 && \text{weights (coefficients) } \langle 1,\ 1,\ 1,\ 1\rangle \\ r(x) &= 4x^3 - 3x && \text{weights (coefficients) } \langle 0,\ -3,\ 0,\ 4\rangle \end{aligned}$$

In general, a polynomial of degree $(n-1)$ with coefficients (weights) $\langle a_0, a_1, \ldots, a_{n-1}\rangle$ can be written as

$$p(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_2x^2 + a_1x + a_0$$

Polynomials can be used to approximate more complex functions. For this course, their primary importance is in modeling positional number systems. From your prior study of mathematics, you should know about low degree polynomials. In particular, their graphs and solutions to their equations.

See the Weierstrass Approximation Theorem.

A natural number n is usually written in decimal positional notation. For instance,

$$31415 = 3{\cdot}10^4 + 1{\cdot}10^3 + 4{\cdot}10^2 + 1{\cdot}10^1 + 5{\cdot}10^0$$

This is the value of

$$p(x) = 3x^4 + x^3 + 4x^2 + x + 5 \quad \text{at } x = 10.$$

Graphs: You should be able to draw graphs of low degree polynomials.

- Constant polynomials: $y = c$
- Linear polynomials: $y = mx + b$
- Quadratic polynomials: $y = ax^2 + bx + c$

Polynomial equations: You should be able to compute solutions low degree polynomial equations.

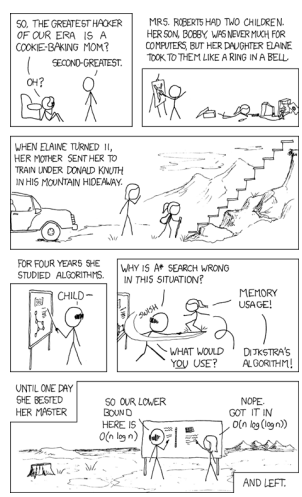
- Linear equations: $mx + b = 0$ has solution $x = -m^{-1}b$
- Quadratic equations: $ax^2 + bx + c = 0$ has solutions

$$x = (2a)^{-1}(-b \pm \sqrt{b^2 - 4ac})$$

Logarithms and exponentials

LOGARITHMS SIMPLIFY CALCULATIONS. To make calculations needed in astronomy for use in navigation, reckoning time, and other applications, logarithms were invented by John Napier and others.

Following Knuth's notation, the binary logarithm is written $\lg x$, the base 2 logarithm of x. The inverse of $\lg x$ is the exponential function 2^x. There are three basic logarithms, but each is just a constant times the others.



1337: Part 2

The old calculus joke relating logarithms and integrals is

$$\int \frac{1}{\text{cabin}}\, d\text{cabin} = \log \text{cabin} + C = \text{houseboat}$$

1. The *common* logarithm, the logarithm base 10, written as a function $y = \log x$ which means $10^y = x$.
2. The *natural* logarithm, the logarithm base $e \approx 2.718$, written as a function $y = \ln x$ which means $e^y = x$.
3. The *binary* logarithm, the logarithm base 2, written as a function $y = \lg x$ which means $2^y = x$.

Logarithms and exponentials have many applications. The study of algorithmic *complexity* is just one use.

From your prior study of mathematics, you should know:

- A log of a product is the sum of logs

$$\log(ab) = \log a + \log b$$

- A log of a quotient is the difference of logs

$$\log\left(\frac{a}{b}\right) = \log a - \log b$$

- A log of a power is the power times the log

$$\log\left(a^b\right) = b \log a$$

- A log of a root is the log divided by the root

$$\log\left(\sqrt[b]{a}\right) = \frac{\log a}{b}$$

- Change of base rule for logarithms

$$\log_b a = \frac{\log_c a}{\log_c b} \quad \text{or} \quad \log_b a \log_c b = \log_c a$$

$$\lg x = \lg 10 \log x \approx 3.322 \log x$$
$$\log x = \log 2 \lg x \approx 0.301 \lg x$$
$$\ln x = \ln 2 \lg x \approx 0.693 \lg x$$

- Exponential–Logarithmic symmetry

$$a^{\log_b(c)} = c^{\log_b(a)}$$

Exponential obey related laws.

- An exp of a sum is the product of exps

$$x^{a+b} = x^a x^b$$

- An exp of a difference is the quotient of exps

$$x^{a-b} = \frac{x^a}{x^b}$$

- An exp of product is an exp of an exp

$$x^{ab} = (x^a)^b$$

- The root of an exp is the exp of a quotient

$$\sqrt[b]{x^a} = x^{\frac{a}{b}}$$

- The change of base rule for exponentials

$$b^x = \left(c^{\log_c b}\right)^x$$

$$2^x = \left(10^{\log 2}\right)^x$$
$$10^x = \left(2^{\lg 10}\right)^x$$
$$e^x = \left(2^{\lg e}\right)^x$$

The *change of base rules* are important enough to be named theorems.

Theorem 7: Logarithm: Change of base

Let x, b, c be real positive real numbers with $b \neq 1$ and $c \neq 1$.

$$\log_b x = \frac{\log_c x}{\log_c b}$$

Proof: Logarithm: Change of base

First, rename $\log_b x$ *by saying* $y = \log_b x$. *Then, by definition of logs and exps,* $b^y = x$.
Take the log base c *of both sides of the equation* $b^y = x$.

$$\boxed{\begin{aligned} b^y &= x \\ \log_c b^y &= \log_c x \\ y \log_c b &= \log_c x \\ y &= \frac{\log_c x}{\log_c b} \\ \log_b x &= \frac{\log_c x}{\log_c b} \end{aligned}}$$

Theorem 8: Exponential: Change of base

Let x, b, c be real positive real numbers with $b \neq 1$ and $c \neq 1$.

$$b^x = c^{x \log_c b}$$

Proof: Exponential: Change of base

First, rename b^x *by saying* $c^y = b^x$.

Now take the log base c of both sides of the equation $c^y = b^x$.

$$\begin{aligned} c^y &= b^x \\ y &= x \log_c b \\ c^y &= c^{x \log_c b} \\ b^x &= c^{x \log_c b} \end{aligned}$$

Permutations

$n!$ read "n factorial," is the product of the first n positive integers.

$0! = 1,\ 1! = 1,\ 2! = 2,\ 3! = 6,\ 4! = 24, \ldots$

$\binom{n}{m}$ is read "n choose m." For small values a binomial coefficient can be evaluated using factorials.

$$\binom{n}{m} = \frac{n!}{m!(n-m)!}$$

PERMUTATIONS ARRANGE THINGS BASED ON SOME CRITERION. Permutations are discussed in more detail in the notes on sorting. Permutations are functions that map a set in a one-to-one manner onto itself.

$$\mathbb{PERMUTATION} = \{\pi(x) : \pi : \mathbb{X} \to \mathbb{X} \text{ is a a one-to-one and onto function}\}$$

A basic fact is the relation (function) between the cardinality of $\mathbb{X}$ and the number of permutations. There are $n!$ permutations on n different things. More generally, there are

$$\begin{aligned} \frac{n!}{(n-m)!} &= n(n-1)(n-2)\cdots(n-m+1) \\ &= m!\binom{n}{m} \end{aligned}$$

ways to *choose* m of n things and then arrange the m things. Factorials are discussed in the notes on recursion. Binomial coefficients are discussed in the notes on counting.

Integer functions

INTEGER FUNCTION ACT ON OR PRODUCE INTEGER RESULTS. There are several useful integer functions: The floor, ceiling, quotient, remainder, and greatest common divisor functions are some of these.

This notation is used for these functions.

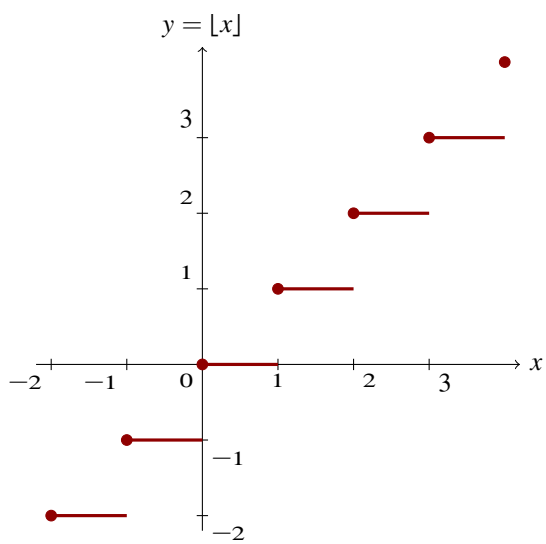
- $\lfloor x \rfloor$ is the floor of x
- $\lceil x \rceil$ is the ceiling of x
- $a \div m$ is the integer quotient when a is divided by m
- $a \bmod m$ is the natural number remainder when a is divided by m
- $\gcd(a, b)$ is the greatest common divisor of a and b.

The floor $\lfloor x \rfloor$ or *greatest integer function* is the largest integer less than or equal to x. For instance,

$$\lfloor \pi \rfloor = 3 \qquad \left\lfloor \frac{24}{5} \right\rfloor = 4 \qquad \lfloor 17 \rfloor = 17$$

$$\lfloor -\pi \rfloor = -4 \qquad \left\lfloor \frac{-24}{5} \right\rfloor = -5 \qquad \lfloor -17 \rfloor = -17$$

Here's a plot of the floor for small values. Such graphs describe *step functions*.

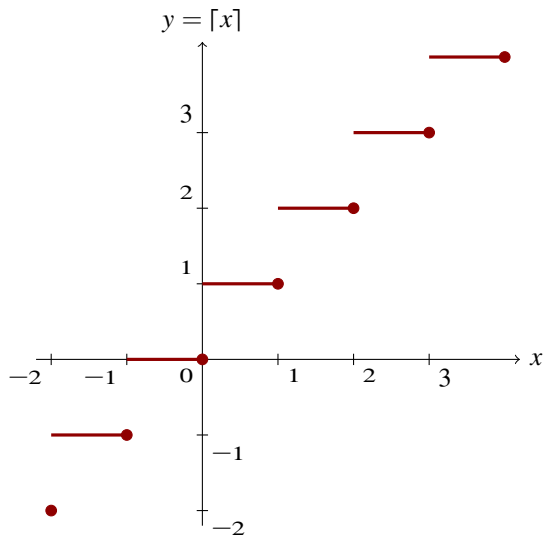


The ceiling $\lceil x \rceil$ or *least integer function* is the smallest integer greater than or equal to x. For instance,

$$\lceil \pi \rceil = 4 \qquad \left\lceil \frac{24}{5} \right\rceil = 5 \qquad \lceil 17 \rceil = 17$$

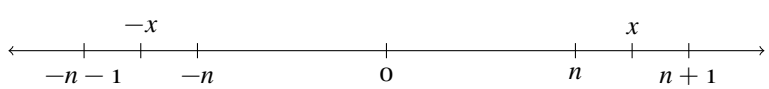
$$\lceil -\pi \rceil = -3 \qquad \left\lceil \frac{-24}{5} \right\rceil = -4 \qquad \lceil -17 \rceil = -17$$

Here's a plot of the ceiling for small values.



Consider x, bounded between by two consecutive integers $n < x < n + 1$.

- The floor of x is $\lfloor x \rfloor = n$
- The ceiling of x is $\lceil x \rceil = n + 1$
- The floor of $-x$ is $\lfloor -x \rfloor = -n - 1$
- The ceiling of $-x$ is $\lceil -x \rceil = -n$



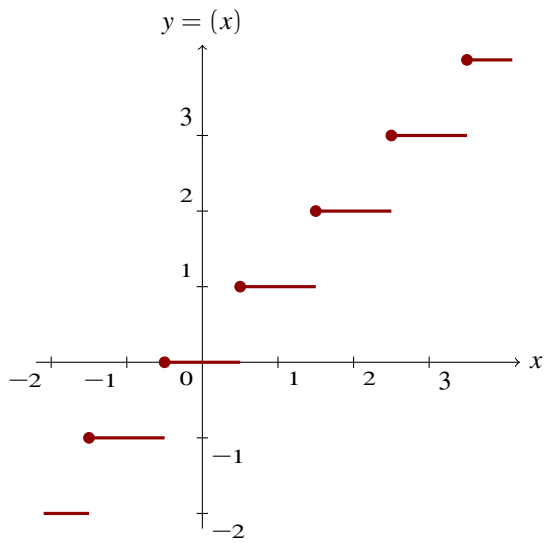
The round (x) or *nearest integer function* is the integer closest to x. In case of a tie at the half-way point $n + 0.5$ between n and $n + 1$, our choice is to always round up to $n + 1$. That is,

$$(x) = \lfloor x + 0.5 \rfloor$$

For instance,

$$(\pi) = 3 \qquad \left(\frac{24}{5}\right) = 5 \qquad (17) = 17$$
$$(-\pi) = -3 \qquad \left(\frac{-24}{5}\right) = -5 \qquad (-17) = -17$$

Here's a plot of the *round* function for small values.



The notation $\{x\}$ for the fractional part of x is unfortunate because it can be confused with the set containing x. But, it is Knuth's notation so I adopt it too.

The fractional part $\{x\}$ or *residue function* is the difference between x and the floor of x.

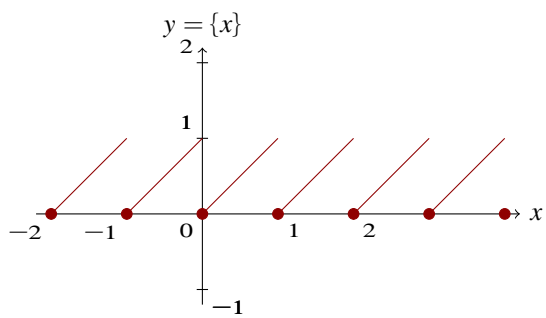
$$\{x\} = x - \lfloor x \rfloor$$

The fraction part start from 0 at an integer n and increases along a line with slope 1 until $(n+1)$ is reached, where the fractional part drops back to 0.

For instance,

$$\{\pi\} \approx 0.1415\cdots \qquad \left\{\frac{24}{5}\right\} = \frac{4}{5} \qquad \{17\} = 0$$

$$\{-\pi\} \approx 0.8584\cdots \qquad \left\{\frac{-24}{5}\right\} = \frac{1}{5} \qquad \{-17\} = 0$$



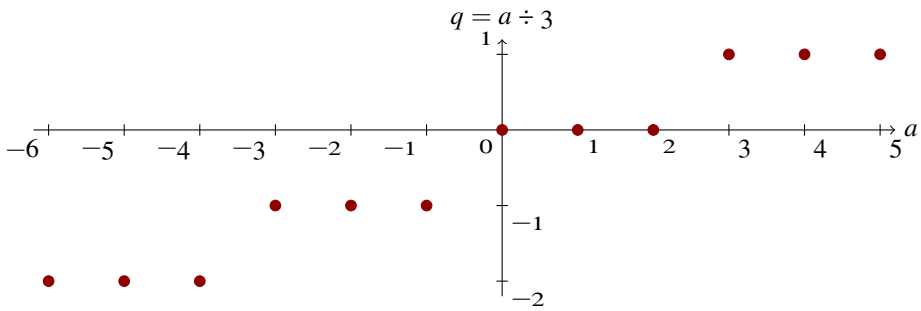
The quotient $q = a \div m$ is the floor of a/m. Here, the input values a and m are integers with $m \neq 0$.

$$q = a \div m = \left\lfloor \frac{a}{m} \right\rfloor \quad \text{is the quotient of } a \text{ divided by } m.$$

For instance,

$5 \div 3 = 1$	$24 \div 5 = 4$	$17 \div 1 = 17$
$-5 \div 3 = -2$	$-24 \div 5 = -5$	$-17 \div 1 = -17$

The integer quotient remains constant at value q for $a = mq$, $mq+1, \ldots, mq+(m-1)$, where it jumps to $(q+1)$ when a reaches $(m+1)q$. Here's a plot for $m = 3$ and small values of a.



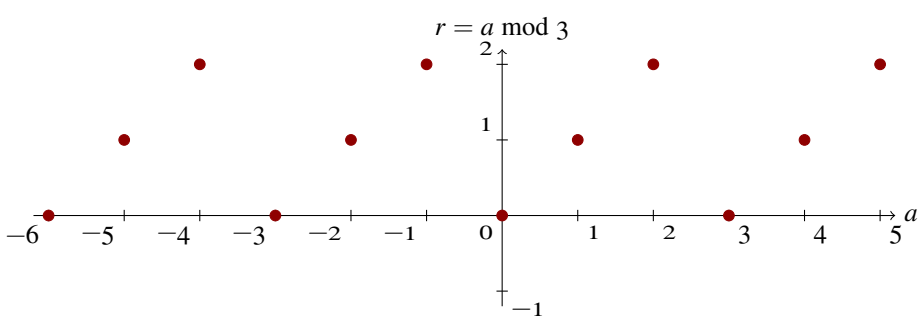
The remainder $r = a \bmod m$ is the remainder when a is divide by m.

$r = a - mq = a - m(a \div m) = a - m\left\lfloor \frac{a}{m} \right\rfloor$ is the remainder of a divided by m.

For instance,

$5 \bmod 3 = 2$	$24 \bmod 5 = 4$	$17 \bmod 1 = 0$
$-5 \bmod 3 = 1$	$-24 \bmod 5 = 1$	$-17 \bmod 1 = 0$

The remainder plot starts at $r = 0$ when a is a multiple of m. The remainder steps by 1 as a increments until it reaches the next multiple of m, where it returns to 0. Here's a plot for $r = a \bmod 3$ and small values of a.

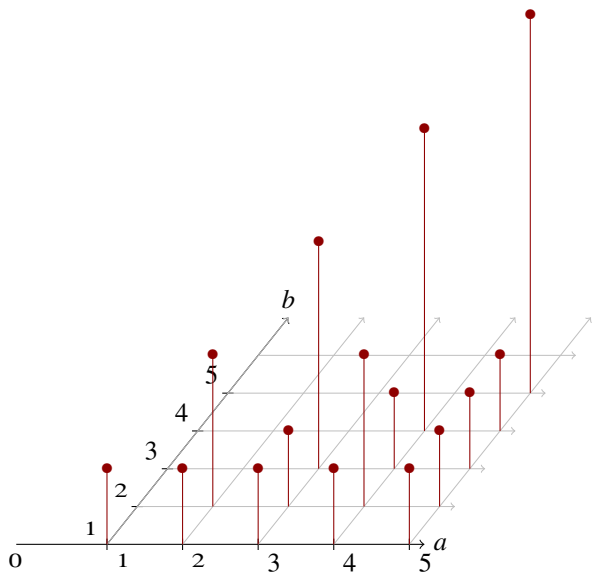


The greatest common divisor, $\gcd(a, b)$ is the largest positive integer that divides both a and b.

For instance,

$\gcd(3, 5) = 1$	$\gcd(5, 24) = 1$	$\gcd(73, 37) = 1$
$\gcd(6, 15) = 3$	$\gcd(25, 35) = 7$	$\gcd(30, 42) = 6$

The greatest common divisor can be pictured as a (3-dimensional) height-field above points (a, b).



Pass a Quiz: Examples of Functions

Take a quiz on page 314 to check your understanding. You can return to here from the quiz.

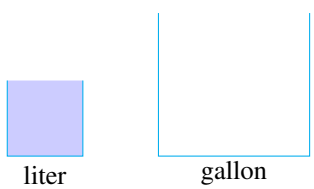
Conversions among units of measure

How do you convert liters into gallons? Feet into meters? Pounds into grams? And other measures from one system to another? Consider the *experimental* method described below that computes these conversion factors is by repeated approximations using *modular* real numbers.

Suppose you had a glass beaker that held exact 1 liter. Also suppose you had an empty jar that held exactly 1 gallon.

Well you *Google* the question, don't you?

This practical example of conversion between units has the Euclidean algorithm as its discrete analogy a sequence of quotient and remainder operations are used to compute the greatest common divisor.

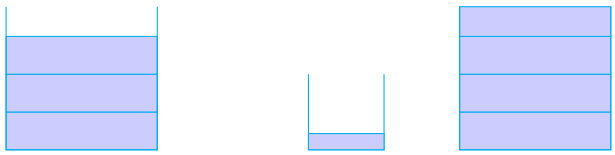


Consider this experiment:

1. Dump the contents of a completed filled beaker into the gallon jar.
2. Repeat step 1 as many times as possible without overfilling the jug.

You would find 3 liters can fit in gallon jar, but 4 cannot. When the fourth liter is poured there will be a *residue* left in the glass beaker.

$$3.0 \text{ liters} < 1 \text{ gallon} < 4.0 \text{ liters}$$



Since the residue, call it r, is less than $1/2$ of the beaker's volume, one 1 gallon is somewhere between 3.5 and 4 liters.

Repeat the process you can get a better estimate. That is, suppose you fabricate a beaker that holds exactly the residue from the fourth liter, and you use it to fill a liter bottle: It goes in 4 times without spilling. The fifth pour fills the liter and there is some small *secondary residue*.

Therefore, the first residue is more than one-fifth and less than one-fourth of a liter. That is, more than three-quarters and less than four-fifths of the fourth liter is needed to fill a gallon.

$$(0.2 < r < 0.25) \Rightarrow (3.75 \text{ liters} < 1 \text{ gallon} < 3.8 \text{ liters})$$

You can repeatedly apply this quotient (integer part) and remainder (residue, fractional part) process and converge on the conversion factor

$$1 \text{ gallon} \approx 3.78541 \text{ liters}.$$

Sets and functions among them are an example of a more general concept called *categories*. Category theory is useful in computer science.

Homework Questions

Use your time outside of class to solve these problems.

1 Let $|\mathbb{X}| > |\mathbb{Y}|$. Determine if the statements below are `True` or `False`: Justify your answer.

1.1 Every function $f : \mathbb{X} \to \mathbb{Y}$ is onto.

1.2 No function $f : \mathbb{X} \to \mathbb{Y}$ can be onto.

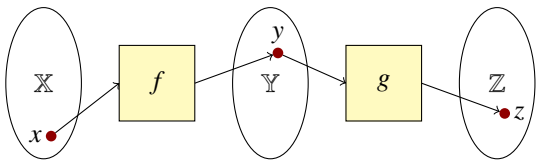
1.3 Some function $f : \mathbb{X} \to \mathbb{Y}$ will be onto.

1.4 Some function $f : \mathbb{X} \to \mathbb{Y}$ will not be onto.

1.5 Answer the same questions when $|\mathbb{X}| < |\mathbb{Y}|$.

2 Composition is an operation on functions. Let $f : \mathbb{X} \to \mathbb{Y}$ and $g : \mathbb{Y} \to \mathbb{Z}$. The *composition* $g \circ f$ maps $\mathbb{X}$ to $\mathbb{Z}$ by the rule

$$(g \circ f)(x) = g(f(x))$$



2.1 If $g(x) = \lg x$ and $f(x) = 2^x$ what is $(g \circ f)(x)$? Simplify your answer.

2.2 If $g(x) = 2^x$ and $f(x) = lgx$ what is $(g \circ f)(x)$? Simplify your answer.

2.3 If $g(x) = x^2$ and $f(x) = 4x - 3$ what is $(g \circ f)(x)$? Simplify your answer.

2.4 If $g(x) = 4x - 3$ and $f(x) = x^2$ what is $(g \circ f)(x)$? Simplify your answer.

3 When I looked (2014-06-25) there were about $7,242,636,000$ billion people on Earth. That is, the world's population was between 232 and 233 billion. Recall IPv4 addresses are 32-bits (4-bytes) wide. I've just summed the number of reserved IPv4 addresses and came up with $592,643,079$.

3.1 If there are 2^{33} people on earth and each is assigned an IPv4 address, Show that some address must assigned be assigned to at least 3 people.

3.2 If there are 2^{33} people on earth and each is assigned an IPv4 address, Show that 3 people cannot be assigned to each address.

4 This is the change of base formula for logarithms. Let $\lg x = y$. Write $log_4 x$, $\log_8 x$, $\log_{16} x$, and $\log x$ in terms of y.

5 Show that $a^{\log_b c} = c^{\log_b a}$.

6 Show that $b = x^{\frac{1}{\log_b x}}$.

7 Use the "sum of logs is log of a product rule" to write the sum

$$\sum_{1 \le k \le n} \lg k$$

as a logarithm that involves a factorial.

8 I found this question in (Graham et al., 1989). Given that $\ln 2 \approx 0.693147$ and $\log 2 \approx 0.301030$, show that

$$\lg x \approx \ln x + \log x$$

Specifically, that the relative error is less than 1% in that

$$\left| \frac{\ln x + \log x}{\lg x} - 1 \right| < 0.01$$

9 Pretend you and your assistant are showing a card trick to friends. You deal your assistant a five card hand:

$$(A\spadesuit),\ (9,\ \diamondsuit),\ (2,\ \clubsuit),\ (5,\ \clubsuit),\ (Q,\ \heartsuit)$$

Your assistant will show you four cards and you will know the fifth hidden card. The first step is to have your assistant code the suit of the hidden card by the arrangement of the revealed cards. How can this be done?

10 Shannon defined the *information content* $I(x)$ of an event x that occurs with probability $0 < p \leq 1$ to be

$$I(x) = \lg 1/p = -\lg p \quad \text{bits.}$$

The average information content, or *entropy*, of a sequence $\vec{S}$ of events $x_0, x_1, x_2, \ldots, x_{n-1}$ with corresponding probabilities $p_0, p_1, p_2, \ldots, p_{n-1}$ is defined to be

$$H(\vec{S}) = -\sum_{k=0}^{n-1} p_k \lg p_k$$

10.1 When flipping a fair coin the probability of heads is $p_H = 0.5$ and probability of a tails is $p_T = 0.5$. What is the information content of flipping a single heads? What is the entropy of flipping the sequence heads, tails, heads?

10.2 When tossing a pair of fair dice the probability of rolling a total of 2 is $p_2 = 1/36$, and the probability of rolling a total of 7 is $p_7 = 6/36 = 1/6$, and the probability of rolling a total of 11 is $p_{11} = 2/36 = 1/18$. What is the information content of rolling a total of 2? A 7? An 11? What is the entropy of rolling the sequence of totals $\langle 2, 7, 11 \rangle$?

11 Hash tables are used in search and other applications. Suppose you are given a hash function

$$h : \text{WORDS} \to \text{NUMBERS}$$

that turns a word into a number, an index into the hash table, where the word will be stored. Hashing can reduce `search` to constant time, a good thing. But collisions occur when two different words hash to the same number.

Suppose you define a 200 element hash table, and you start inserting words.

11.1 At what point do you know a collision must have occurred.

11.2 At what point do you know three or more words must have hashed to the same location?

12 A recent commercial has advertised that, in summer, in Florida, after setting your thermostat to 78°, you will save 6% on your electric bill each time you raise the setting by 1°. At what temperature should you set your thermostat to get your electricity for free?

13 I think the interpretation in question 12 may be wrong. That model is *simple* deductions.

I suspect the electric company meant *compound* deductions at 6%. Using this model at what thermostat will your electricity be free?

6. Horner's Rule: Polynomials & number conversions

The best number is 73. Why? 73 is the 21st prime number. Its mirror (37) is the 12th and its mirror (21) is the product of multiplying 7 and 3. In binary, 73 is a palindrome, 1001001 which backwards is 1001001.

Dr. Sheldon Cooper (Jim Parsons) in The Big Bang Theory, Season 4 Episode 10
"The Alien Parasite Hypothesis"

Polynomials

Polynomials are a useful class of functions for several reasons. They can be:

Although this method is named for Horner, it was known by Qin Jiushao about 600 years before Horner.

- *Evaluated* efficiently: By Horner's rule, called *synthetic division* in high-school algebra.
- *Differentiated* easily: $dx^n/dx = nx^{n-1}$. A concept studied in calculus as a continuous extension of differences.
- *Integrated* easily: $\int x^n\,dx = \frac{x^{n+1}}{n+1} + c$. A concept studied in calculus as a continuous extension of summations.

And, perhaps most importantly, polynomials can

- *Approximate* many other functions arbitrarily well under simple restrictions.

See the Weierstrass Approximation Theorem.

The set of *power functions* is the *standard basis* used for polynomials.

$$\boxed{\{x^n : n \in \mathbb{N}\} = \{x^n : n = 0,\ 1,\ 2,\ 3, \ldots\} = \{1,\ x,\ x^2,\ x^3, \ldots\}}$$

Written in the power basis, a polynomial $p(x)$ has the form

$$\boxed{p(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_2x^2 + a_1x + a_0} \tag{6}$$

It is interesting to consider a polynomial as the inner product of a row of coefficients with a column of powers

$$\begin{bmatrix} a_0 & a_1 & \cdots & a_{n-1} \end{bmatrix} \begin{bmatrix} 1 \\ x \\ \vdots \\ x^{n-1} \end{bmatrix}$$

where the *degree* of $p(x)$ in equation (6) is $(n-1)$. and the *coefficients* are $\langle a_0,\ a_1, \ldots,\ a_{n-2},\ a_{n-1}\rangle$ A *root* or *zero* of $p(x)$ is a *value of* x such that

$$\boxed{p(x) = 0} \tag{7}$$

The fundamental theorem of algebra states that every polynomial of degree $(n-1) > 0$ has $(n-1)$ roots, some may be complex numbers and some may occur more than once.

Root-finding is a fundamental numerical analysis tasks and there exist many algorithms that search for a polynomial's roots. Newton's method is mentioned in these notes.

Theorem 9: Fundamental theorem of algebra

Every single-variable polynomial of degree $(n-1) > 0$ with real coefficients has exactly $(n-1)$ roots, counted by multiplicity.

Falling factorial powers

In the sum-and-difference calculus, *falling factorial* powers will provide a natural basis for writing polynomials. The *falling factorial* are:

Definition 14: Falling factorial powers

Falling factorial powers are defined by:

- *A base (initial) case:* $x^{\underline{0}} = 1$.
- *Recursion rule:* $x^{\underline{n}} = x^{\underline{n-1}} \cdot (x - (n-1))$.

For instance,

$$\begin{aligned}
x^{\underline{0}} &= 1 \\
x^{\underline{1}} &= x^{\underline{0}}(x-0) = x \\
x^{\underline{2}} &= x^{\underline{1}}(x-1) = x(x-1) \\
x^{\underline{3}} &= x^{\underline{2}}(x-2) = x(x-1)(x-2) \\
\vdots &= \quad \vdots \\
x^{\underline{n}} &= x^{\underline{n-1}}(x-(n-1)) = x(x-1)\cdots(x-(n-1))
\end{aligned}$$

An fundamental result of calculus is

$$\text{The } \textit{derivative} \text{ of } \quad x^n \quad \text{is} \quad \frac{d}{dx}x^n = nx^{n-1}$$

Note that

$$\frac{(x+h)^n - x^n}{h} = \frac{\left(x^n + \binom{n}{1}x^{n-1}h + \cdots + \binom{n}{n}h^n\right) - x^n}{h} = nx^{n-1} + o(h)$$

So that the divided difference approaches nx^{n-1} as $h \to 0$.

The discrete analogy is:

$$\text{The } \textit{difference} \text{ of } \quad x^{\underline{n}} \quad \text{is} \quad \triangle x^{\underline{n}} = nx^{\underline{n-1}}$$

The proof is algebraic manipulation.

$$\begin{aligned}
\triangle x^{\underline{n}} &= (x+1)^{\underline{n}} - x^{\underline{n}} \\
&= \left[(x+1)x^{\underline{(n-1)}}\right] - \left[x^{\underline{(n-1)}}\right](x-(n-1)) \\
&= [(x+1) - (x-(n-1))]\, x^{\underline{(n-1)}} \\
&= nx^{\underline{n-1}}
\end{aligned}$$

The *falling factorial powers* are related to binomial coefficients by the identity.

$$x^{\underline{n}} = x(x-1)\cdots(x-n+1) = \frac{x!}{(x-n)!} = n!\binom{x}{n}$$

Do you find it interesting that these formalizes are valid even when x is not a natural number?

The falling factorial powers are a natural basis for the In the sum-and-difference calculus. The fundamental theorem of the sum-and-difference calculus is the equation

$$\sum_{0\le x<n} x^{\underline{n}} = \frac{x^{\underline{n+1}}}{n+1}$$

This follows from the fact that the sum of values in column n of Pascal's triangle up to row x is equal to the value in the column $(n+1)$ and row $(x+1)$. That is

$$\boxed{\sum_{0\le x<n} x^{\underline{n}} = n!\sum_{0\le x<n}\binom{x}{n} = n!\binom{x+1}{n+1} = \frac{(x+1)^{\underline{n+1}}}{(n+1)!}}$$

Stirling's numbers of the second kind relate the polynomial power basis to *falling factorial* powers. Consider, the following expansions.

$$\boxed{\begin{aligned}
x^0 &= x^{\underline{0}} = 1\\
x^1 &= x^{\underline{1}} = x\\
x^2 &= x^{\underline{2}} + x^{\underline{1}} = x(x-1) + x\\
x^3 &= x^{\underline{3}} + 3x^{\underline{2}} + x^{\underline{1}}\\
&= x(x-1)(x-2) + 3x(x-1) + x\\
x^4 &= x^{\underline{4}} + 6x^{\underline{3}} + 7x^{\underline{2}} + x^{\underline{1}}\\
&= x(x-1)(x-2)(x-3) + 6x(x-1)(x-2) + 7(x-1) + x
\end{aligned}}$$

The general formula is

$$\boxed{x^{n-1} = \sum_k \left\{ {n-1 \atop k} \right\} x^{\underline{k}}} \tag{8}$$

Where $\left\{ {n \atop k} \right\}$ are Stirling's numbers of the second kind, that are defined by the recurrence

$$\left\{ {n \atop k} \right\} = k\left\{ {n-1 \atop k} \right\} + \left\{ {n-1 \atop k-1} \right\}$$

Notice that *falling factorial powers* satisfy the recurrence

$$x^{\underline{k+1}} = x^{\underline{k}}(x-k) \quad \text{or} \quad x\cdot x^{\underline{k}} = x^{\underline{k+1}} + kx^{\underline{k}}$$

Therefore, if equation (8) is valid at $(n-1)$, then

$$\begin{aligned}
x^n &= x \cdot x^{n-1} \\
&= \sum_k \left\{ {n-1 \atop k} \right\} x \cdot x^{\underline{k}} \\
&= \sum_k \left\{ {n-1 \atop k} \right\} \left(x^{\underline{k+1}} + k x^{\underline{k}} \right) \\
&= \sum_k \left\{ {n-1 \atop k} \right\} x^{\underline{k+1}} + \sum_k k \left\{ {n-1 \atop k} \right\} x^{\underline{k}} \\
&= \sum_k \left\{ {n-1 \atop k-1} \right\} x^{\underline{k}} + \sum_k k \left\{ {n-1 \atop k} \right\} x^{\underline{k}} \\
&= \sum_k \left(\left\{ {n-1 \atop k-1} \right\} + k \left\{ {n-1 \atop k} \right\} \right) x^{\underline{k}} \\
&= \sum_k \left\{ {n \atop k} \right\} x^{\underline{k}}
\end{aligned}$$

There are many useful polynomial bases. Some classical polynomials are: Chebyshev, Hermite, Jacobi, Lagrange, and Legendre polynomials. The choice of how to represent a polynomial depends on the application.

See (Graham et al., 1989) for a full discussion of the relationships among *powers* x^n, *falling factorial powers* $x^{\underline{n}}$ and *rising factorial powers* $x^{\overline{n}}$.

Give references for these.

Evaluating a polynomial

Consider the golden polynomial $g(x) = x^2 - x - 1$, and suppose you want to evaluate $g(x)$ at $x = 3$. By brute force you compute

$$g(3) = 3 \cdot 3 - 3 - 1 = 5$$

taking 1 multiply and 2 additions. For the general quadratic, by brute force, you evaluate

$$p(x) = ax^2 + bx + c = a \cdot x \cdot x + b \cdot x + c$$

taking 3 multiplies and 2 additions.

Horner's rule computes the value of the polynomial from the highest exponent to the lowest.

$$p(x) = ((a \cdot x) + b) \cdot x + c$$

which requires only 2 multiplies and 2 additions. Although this may not seem to be a significant improvement, it is: Multiplies increase quadratically in the brute force algorithm, but only linearly in Horner's Rule. For instance, naively evaluating

$$p(x) = ax^3 + bx^2 + cx + d = a \cdot x \cdot x \cdot x + b \cdot x \cdot x + c \cdot x + d$$

requires 6 multiplies and 3 additions, but

$$p(x) = (((a \cdot x + b) \cdot x) + c) \cdot x + d$$

only requires 3 multiplies and 3 additions.

Here's an example. Let the polynomial be

$$p(x) = 2x^4 - 5x^3 + 3x^2 + 5x - 7$$

and pretend you need its value at, say $x = -3$. That is, you need to compute the value of $p(-3)$. Horner rule, sometime called synthetic division, is an efficient way to make this computation.

The first step is to copy the coefficients of

$$p(x) = 2x^4 - 5x^3 + 3x^2 + 5x - 7$$

leaving sufficient space to make some calculations Then leave a blank line and draw a totaling line underneath it.

Horner's rule (synthetic division)

HORNER'S RULE @ $x = -3$

2	−5	3	5	−7	Coefficients

The value of $p(x)$ at $x = -3$ is 302: $p(-3) = 302$. This can be used to infer that when

$$p(x) = 2x^4 - 5x^3 + 3x^2 + 5x - 7$$

is divided by $(x + 3)$ you will compute a quotient polynomial

$$q(x) = 2x^3 - 11x^2 + 36x - 103 \quad \text{and a remainder of} \quad r = 302$$

$$2x^4 - 5x^3 + 3x^2 + 5x - 7 = (2x^3 - 11x^2 + 36x - 103)(x + 3) + 302$$

Example: Horner's rule

Evaluate $p(x) = x^3 - x^2 - x - 1$ *at* $x = 4$.

Horner's Rule @ $x = 4$

1	−1	−1	−1
	4	12	44
1	3	11	$\boxed{43}$

Evaluate $p(x) = 7x^5 - 5x^3 + 3x^2$ *at* $x = -2$.

Horner's Rule @ $x = -2$

7	0	−5	3	0	0
	−14	28	−46	86	−172
7	−14	23	−43	86	$\boxed{-172}$

```
horner :: (Num a) => a -> [a] -> a -- a number (value of x)
                                   -- and list of numbers (coefficients)
                                   -- maps to a number
horner x [] = 0
horner x [c] = c
horner x [c:d:cs] = horner x [c*x+d:cs]
```

Pass a Quiz: Horner's rule

Take a quiz on page 315 to check your understanding. You can return to here from the quiz.

Explain how to iterate Horner's rule to compute Taylor coefficients

Conversion to decimal

CONVENTIONALLY, NUMBERS ARE WRITTEN IN POSITIONAL NOTATION. For instance, writing 314 states 3 hundreds, 1 ten, and 4 ones make the number 314.

$$314 = 3 \cdot 10^2 + 1 \cdot 10^1 + 4 \cdot 10^0$$

Fixed-point numbers are also written in positional notation. For instance, writing 2.718 states 2 ones, 7 tenths, 1 hundredth, and 8 thousandths make 2.718.

$$2.718 = 2 \cdot 10^0 + 7 \cdot 10^{-1} + 1 \cdot 10^{-2} + 8 \cdot 10^{-3}$$

Numbers in bases other that 10 are written the similarly: Only the set of valid coefficients and the base differ.

Binary notation

Binary numbers are written using bits, elements from $\mathbb{B} = \{0, 1\}$, and powers of 2. For instance,

$$\begin{aligned}
(1101)_2 &= 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \\
&= 13 \\
(1001001)_2 &= 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \\
&= 73 \\
(100.1001)_2 &= 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} \\
&= \frac{73}{16}
\end{aligned}$$

Octal notation

Octal numbers are written using octal numerals, elements from $\mathbb{O} = \{0, 1, 2, 3, 4, 5, 6, 7\}$, and powers of 8. For instance,

In "The Shining," a famous film by Stanley Kubrick staring Jack Nicholson, the little boy, Danny, is warned to stay out of room 237.

There is an old joke: `Halloween = Christmas` because `OCT 31 = DEC 25`.

$$\begin{aligned}
(237)_8 &= 2 \cdot 8^2 + 3 \cdot 8^1 + 7 \cdot 8^0 = 159 \\
(7.3)_8 &= 7 \cdot 8^0 + 3 \cdot 8^{-1} = \frac{59}{8}
\end{aligned}$$

Hexadecimal notation

Hexadecimal numbers are written using hexadecimal numerals, elements from $\mathbb{H} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$, and powers of 16. In this case, the letters A through F stand for two-digit decimal numbers:

$$A = 10, \quad B = 11, \quad C = 12, \quad D = 13, \quad E = 14, \quad F = 15$$

For instance,

$$\begin{aligned}
(DEAD)_{16} &= 13 \cdot 16^3 + 14 \cdot 16^2 + 10 \cdot 16^1 + 13 \cdot 16^0 \\
&= 57,005 \\
(C0.DE)_{16} &= 12 \cdot 16^1 + 0 \cdot 16^0 + 13 \cdot 16^{-1} + 14 \cdot 16^{-2} \\
&= \frac{49374}{256} = 192.8671875
\end{aligned}$$

Positional notation is polynomial notation

In each of the cases above, a number can be thought of as a polynomial evaluated at a given base: 10, 2, 8, or 16. The coefficients in each case is restricted to a particular alphabet: $\mathbb{D}$, $\mathbb{B}$, $\mathbb{O}$, or $\mathbb{H}$. Therefore, Horner's rule can be used to convert from some base to decimal. Here are some examples.

- Convert the binary number $(100\ 1001)_2$ to decimal.

Horner's Rule @ $x = 2$

1	0	0	1	0	0	1
	2	4	8	18	36	72
1	2	4	9	18	36	$\boxed{73}$

Therefore $(100\ 1001)_2 = 73$.

- Convert the binary number $(0010\ 1010)_2$ to decimal.

Horner's Rule @ $x = 2$

0	0	1	0	1	0	1	0
	0	0	2	4	10	20	42
0	0	1	2	5	10	21	$\boxed{42}$

Therefore $(0010\ 1010)_2 = 42$.

- Convert the binary number $(1100\ 1100)_2$ to decimal.

Horner's Rule @ $x = 2$

1	1	0	0	1	1	0	0
	2	6	12	24	50	102	204
1	3	6	12	25	51	102	$\boxed{204}$

Therefore $(1100\ 1100)_2 = 204$.

- Convert the octal number $(237)_8$ to decimal.

Horner's Rule @ $x = 8$

2	3	7
	16	152
2	19	$\boxed{159}$

Therefore $(237)_8 = 159$.

Fixed-point to decimal

Horner's rule can be used to convert fixed-point numbers, written in some base, to decimal. You know that 3.1415 can be written as a fraction

$$3.1415 = \frac{31415}{10000} = \frac{31415}{10^4}$$

In a similar fashion

$$(101.1101)_2 = \frac{(1011101)_2}{16} = \frac{(1011101)_2}{2^4}$$

Therefore you can compute the decimal value of a fixed-point number, by (1) ignoring the point, (2) executing Horner's algorithm on the number, and (3) dividing by the *appropriate power* of the base. For instance, to convert $(101.1101)_2$ use Horner's rule

The *appropriate power* is the number of places to the right of the point.

Horner's Rule @ $x = 2$

1	0	1	1	1	0	1
	2	4	10	22	46	92
1	2	5	11	23	46	$\boxed{93}$

Therefore,

$$(101.1101)_2 = \frac{93}{2^4} = \frac{93}{16} = 5.8125$$

When you multiply $(101.1101)_2$ by 2^4 the *binary point* floats to the right end of the string.

$$2^4\,(101.1101)_2 = (1011101.)_2$$

The integer part of $(101.1101)_2$ is $(101)_2 = 5$ and the fractional part is $(0.1101)_2 = \frac{13}{16} = 0.8125$.

Pass a Quiz: Horner's conversion to decimal

Take a quiz on page 315 to check your understanding. You can return to here from the quiz.

Conversion: Decimal to other basis

By reversing the steps of Horner's rule a decimal number can be converted to binary, octal, hexadecimal on any other base. I call this *repeated remaindering*.

The idea is straightforward. Given a natural number n, its remainder r when divided by 2 is either 0 or 1. The remaindering algorithm repeatedly computes n's remainders bit-by-bit by successively dividing quotients by 2.

The first division decides if n is even ($r = 0$) or odd ($r = 1$). This computes the least-significant, rightmost, bit in the binary string that represents n.

The recursion to convert n to binary goes like this. Let

$$(n)_2 = \left(b_{(m-1)} \cdots b_1 b_0\right)_2 \quad \text{where } m = \lfloor \lg n \rfloor + 1$$

- Let $n_0 = n$ and $b_0 = n_0 \bmod 2$
- For $k = 1, 2, \ldots, (m-1)$ define n_k and b_k by

$$n_k = \left\lfloor \frac{n_{(k-1)}}{2} \right\rfloor \quad \text{and} \quad b_k = n_{(k-1)} \bmod 2$$

Here's an example showing how to convert $n = 161$ to binary. The remainder bits b_k are computed from the least significant b_0 to the most significant $b_{(m-1)}$.

Repeated Remaindering mod 2

QUOTIENTS	161	80	40	20	10	5	2	1
REMAINDERS	1	0	0	0	0	1	0	1

Therefore, $(161)_{10} = (1010\ 0001)_2$. As another example, let's convert $n = 237$ to binary.

Repeated Remaindering mod 2

QUOTIENTS	237	118	59	29	14	7	3	1
REMAINDERS	1	0	1	1	0	1	1	1

Therefore, $(237)_{10} = (1110\ 1101)_2$. And here's an example showing conversion to octal.

Repeated Remaindering mod 8

QUOTIENTS	73	9	1
REMAINDERS	1	1	1

Therefore, $(73)_{10} = (111)_8$. Written in HASKELL the algorithm might look like:

```
decToBin :: Natural -> [Bits]
decToBin 0 = [0]
decToBin 1 = [1]
decToBin n = decToBin (n `div` 2) ++ [n `mod` 2]
```

A function to convert a natural number to a bit string recursively converts $n/2$ to binary and concatenates the quotient with the remainder. The $++$ operator is Concatenation.

$$n = \text{BINARY}\left(\frac{n}{2}\right) + +n \bmod 2$$

For instance,

$$\begin{aligned} 37 &= \text{BINARY}\left(\frac{37}{2}\right) + +(37 \bmod 2) \\ &= \text{BINARY}\,(18) + +1 \\ &= \text{BINARY}\,(9) + +01 \\ &= \text{BINARY}\,(4) + +101 \\ &= \text{BINARY}\,(2) + +0101 \\ &= \text{BINARY}\,(1) + +00101 \\ &= 100101 \end{aligned}$$

Decimal to fixed-point

To convert a decimal fraction, say 0.3 to binary use repeated flooring upon multiplication by 2. Let $0 < f < 1$ be a fraction written in decimal notation. Let

$$f = (b)_2 = (0.b_1 b_2 b_3 \cdots)_2$$

be the binary representation of f.

The recursion to convert f to $(b)_2$ is this:

- Let $f_0 = f$ and $(b)_2 = 0.\ldots$

- For $n = 1, 2, \ldots$ let define f_n and b_n by

$$f_n = \begin{cases} p_{(n-1)} & \text{if } 2 \cdot f_{(n-1)} < 1 \\ p_{(n-1)} - 1 & \text{if } 2 \cdot f_{(n-1)} \geq 1 \end{cases}$$

$$b_n = \begin{cases} 0 & \text{if } 2 \cdot f_{(n-1)} < 1 \\ 1 & \text{if } 2 \cdot f_{(n-1)} \geq 1 \end{cases}$$

For instance, here's how to convert 0.3 to binary notation.

$$\begin{aligned}
0.3 \times 2 = 0.6 \qquad & \text{shows } 0.3 < 0.5 = (0.1)_2 \\
& \therefore 0.3 = (0.0\cdots)_2 \\
0.6 \times 2 = 1.2 \qquad & \text{shows } 0.3 > 0.25 = (0.01)_2 \\
& \therefore 0.3 = (0.01\cdots)_2 \\
0.2 \times 2 = 0.4 \qquad & \text{shows } 0.3 < 0.375 = (0.011)_2 \\
& \therefore 0.3 = (0.010\cdots)_2 \\
0.4 \times 2 = 0.8 \qquad & \text{shows } 0.3 < 0.3125 = (0.0101)_2 \\
& \therefore 0.3 = (0.0100\cdots)_2 \\
0.8 \times 2 = 1.6 \qquad & \text{shows } 0.3 > 0.28125 = (0.01001)_2 \\
& \therefore 0.3 = (0.01001\cdots)_2 \\
\vdots &
\end{aligned}$$

And now the pattern repeats:

$$0.3 = (0.0\ 1001\ 1001\ 1001\cdots)_2$$

By changing the multiplier you can convert to another base. For instance, to convert 0.1 to hexadecimal use 16 as the multiplication factor.

$$\begin{aligned}
0.1 \times 16 = 1.6 \qquad & \text{shows } 0.1 = (0.1\cdots)_{16} \\
0.6 \times 16 = 9.6 \qquad & \text{shows } 0.1 = (0.19\cdots)_{16} \\
\vdots &
\end{aligned}$$

Therefore,

$$0.1 = (0.199\cdots)_{16}$$

If you like *series*, studied in calculus, notice that

$$\begin{aligned}
(0.199\cdots)_{16} &= \frac{1}{16} + \frac{9}{16^2}\left[1 + \frac{1}{16} + \left(\frac{1}{16}\right)^2 + \cdots\right] \\
&= \frac{1}{16} + \frac{9}{16^2}\left[\frac{1}{1-\frac{1}{16}}\right] \\
&= \frac{1}{16} + \frac{9}{16^2}\left[\frac{16}{15}\right] \\
&= \frac{1}{16} + \frac{3}{16\cdot 5} \\
&= \frac{8}{16\cdot 5} \\
&= 0.1
\end{aligned}$$

Pass a Quiz: Remaindering: Convert decimal to another base

Take a quiz on page 316 to check your understanding. You can return to here from the quiz.

Homework Questions

Use your time outside of class to solve these problems.

1 Let $p(x) = 2x^4 - 8x^3 - 6x^2 + 5x - 7$.

1.1 Evaluate $p(3)$ using *brute force*. Count the number of multiplies and adds you make.

1.2 Evaluate $p(3)$ using Horner's rule. Count the number of multiplies and adds you make.

2 Consider a general polynomial of degree $(n-1)$.

$$p(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_2x^2 + a_1x + a_0$$

2.1 Argue that evaluating $p(x)$ by brute-force requires $n(n-1)/2$ multiplications and $(n-1)$ additions.

2.2 Argue that evaluating $p(x)$ by Horner's rule only requires $(n-1)$ multiplications and $(n-1)$ additions.

3 Use Horner's rule to convert the following numbers to decimal notation.

3.1 $(1111\ 1111)_2$

3.2 $(1100\ 1100)_2$

3.3 $(1010\ 1010)_2$

3.4 $(747)_8$

3.5 $(123)_4$

3.6 $(A47)_{16}$

4 Use repeated remaindering to convert the following decimal number to the indicated base.

4.1 76 to binary.

4.2 137 to binary

4.3 177 to binary

4.4 76 to octal.

4.5 137 to hexadecimal

4.6 177 to hexadecimal

5 Use your (correct) answers to problem 3 to convert the following fixed-point numbers to decimal notation.

5.1 $(1111.1111)_2$

5.2 $(11.001100)_2$

5.3 $(1.23)_4$

5.4 $(A.47)_{16}$

6 Convert the following fixed-point decimal numbers to the indicated base.

6.1 1.125 to binary (notice this is 1 plus 1/8)

6.2 7.3 to binary

6.3 7.3 to octal

6.4 7.3 to hexadecimal

7 Look what happens when you create a table of $p(n)$, $\triangle p(n)$, and higher-order differences.

$p(n) = n^2 - n - 1$ and its differences

n	$p(n)$	$\triangle p(n)$	$\triangle^2 p(n)$	$\triangle^3 p(n)$
0	-1	0	2	0
1	-1	2	2	0
2	1	4	2	0
3	5	6	2	0
4	11	8	2	0
5	19	10	2	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
m	$p(m-1) + \triangle p(m-1)$	$2m$	2	0

Use this table and your knowledge of Horner's rule to argue that:

7.1 You can evaluate $p(0) = -1$, $p(1) = -1$, and $p(2) = 1$, at a cost of 6 multiplies and 6 additions.

7.2 Then you can evaluate

$$\triangle p(0) = p(1) - p(0) = 0, \quad \triangle p(1) = p(2) - p(1) = 2, \quad \text{and} \triangle^2 p(0) = \triangle p(1) - \triangle p(0)$$

at a cost of 3 more additions (subtractions)
Now show that you can be compute future values of $p(n)$ for $n = 3, 4, 5, \ldots$ at a cost of just 2 addition.

$$\boxed{\begin{aligned} \triangle p(n-1) &= \triangle p(n-2) + 2 \\ p(n) &= p(n-1) + \triangle p(n-1) \end{aligned}}$$

7.3 What is the cost to evaluate $p(n) = n^2 - n - 1$ at $n = 0, 1, \ldots, m$ using Horner's rule at each value of n?

7.4 What is the cost to evaluate $p(n) = n^2 - n - 1$ at $n = 0, 1, \ldots, m$ using the difference ideas developed above?

7.5 There is a general idea here: On a set of equally spaced numbers, say $n = 0, 1, \ldots, m$, a polynomial can be evaluated using a constant number of multiplications and a linear number of additions. Can you:

- Create another example?
- Prove this?
- Look it up?

Maybe work through the details for $p(x) = x^3 - x^2 - x - 1$ at $x = 0, 1, 2, \ldots$.

8 In calculus it is taught that the exponential function $y = \exp x = e^x$ can be approximated by a Polynomials.

$$e^x \approx 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!}$$

Use Horner's rule on the MacLaurin polynomial of degree 4 to approximate e.

9 Pretend you need to evaluate a polynomial written using *falling factorial* powers. That is, given

$$p(x) = a_{n-1} x^{\underline{n-1}} + a_{n-2} x^{\underline{n-2}} + \cdots + a_2 x^{\underline{2}} + a_1 x^{\underline{1}} + a_0$$

you need to compute $p(c)$ for some value $x = c$. How can you modify Horner's rule to do this?

10 *Polynomial interpolation* is a common approximation method. The problem is to find a polynomial $p(t)$ that *fits* observed data

$$(t_0, y_0), (t_1, y_1), \ldots, (t_{n-1}, y_{n-1})$$

Think of t as a time variable and y as some measured value at time t. For instance, suppose you were given a time series$\langle 0, 0, 1, \ldots \rangle$, measured at times $0, 1, 2, \ldots$. *Lagrange* showed how to construct a polynomial that fits the data. In the example case, let

$$p(t) = 0 \cdot \frac{(t-1)(t-2)}{(0-1)(0-2)} + 0 \cdot \frac{t(t-2)}{(1-0)(1-2)} + 1 \cdot \frac{t(t-1)}{(2-0)(2-1)}$$

10.1 Do you agree that $p(0) = 0$, $p(1) = 1$, and $p(2) = 1$?

10.2 Show that you can reduce $p(t)$ to $t(t-1)/2$.

10.3 What is the name (used in these notes) for sequence generated by $p(t)$ at $t = 0, 1, 2, 3, \ldots$?

More generally, given time steps $t_0, t_1, \ldots, t_{n-1}$ define the *Lagrange basis* polynomials by

$$l_j(t) = \prod_{\substack{0<k<n-1 \\ k \neq j}} \frac{(t - t_k)}{(t_j - t_k)} \quad \text{for } j = 0, 1, \ldots, (n-1)$$

10.1 Show that $l_j(t)$ is a *delta* function. That is,

$$l_j(t_i = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

'Let $\langle y_0, y_1, y_2, \ldots \rangle$ be a time series measured at steps $t_0, t_1, \ldots, t_{n-1}$. The *Lagrange* polynomial that interpolates the time series is

$$L(t) = \sum_{k=0}^{n-1} y_k l_k(t)$$

10.1 Show that $L(t)$ interpolates the data. That is, show that

$$L(t_i) = y_i \quad (\forall i = 0, 1, \ldots, (n-1)).$$

11 In the notes on converting decimal fixed-point numbers to another base you saw that

$$0.3 = (0.0\ 1001\ 1001\ 1001 \cdots)_2$$

11.1 Write the binary expansion on the right as a sum of (negative) powers of 2.

11.2 Factor 2^{-2} out of every other term in your expansion to get a geometric series in powers of 2^{-4}.

11.3 Factor 2^{-5} out of every other term, starting from the second term, to get the same geometric series in powers of 2^{-4}.

11.4 Use ideas about geometric series from calculus to show that the geometric series in powers of 2^{-4} is equal to $16/15$.

11.5 Conclude that the binary expansion of 0.3 is correct.

7. *Sequences*: *The sum & difference calculus*

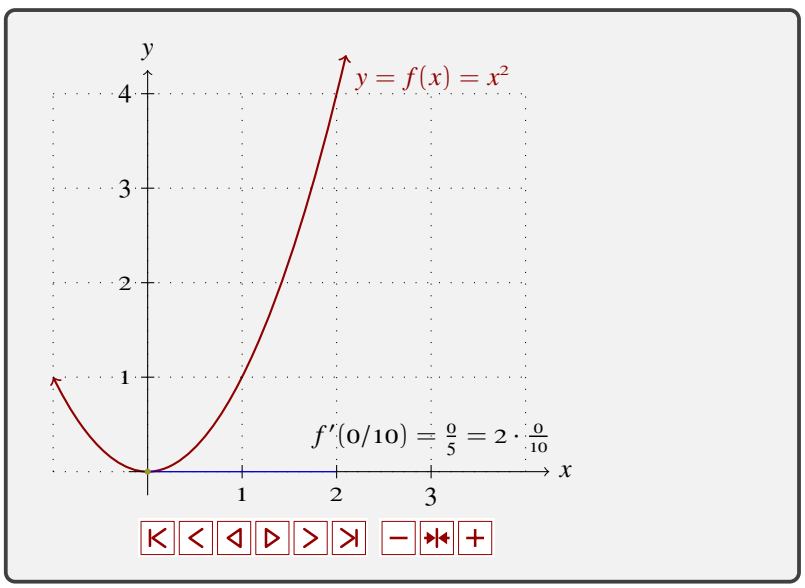
Calculus: *Derivatives and Integrals*

Calculus studies what can be learned from *derivatives* and *integrals* of functions $f : \mathbb{X} \to \mathbb{Y}$. In introductory calculus both the domain $\mathbb{X}$ and the co-domain $\mathbb{Y}$ are subsets of real numbers $\mathbb{R}$.

- The *derivative* f' of a function f at x is the limit of a divided-difference as the interval length h shrinks to zero

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

  The value $f'(x)$ of the derivative is the slope of the line *tangent* to the graph of the curve $y = f(x)$.



- The *integral* is the *area* under the curve and the x-axis, provided $f(x) > 0$. An integral measures summed (accumulated) *values* of function f over a continuous domain.

  A definite (left Riemann) *integral* $\int_a^b f(x)\,dx$ of a function f over an interval $[a, b]$ is the limit of a sum as the number of terms approach

I really don't want you to skip this section! To reconize the analogy between discrete and continuous calculus can be a first step in understanding the role of analogy throughout mathematics and problem solving.

Calculus studies continuous phenomena and is not required for this course. A course requirement is that you be calculus-ready. Some of you have mastered calculus, some have been exposed, and other may have never studied calculus. Most of you will master calculus because you want a career in a field that requires it whether you will use calculus or not. There is a need for people who are willing to think deeply enough to master the intricacies of calculus.

This course mentions calculus topics when discrete structures and algorithms can be used to better understand continuous ideas. Defining the real numbers takes some explanation and thought. Let me just say that a number x is a real number if it is the limit of a Cauchy sequence of rational numbers. See Real Numbers (Wikipedia) and Cauchy sequence (Wikipedia).

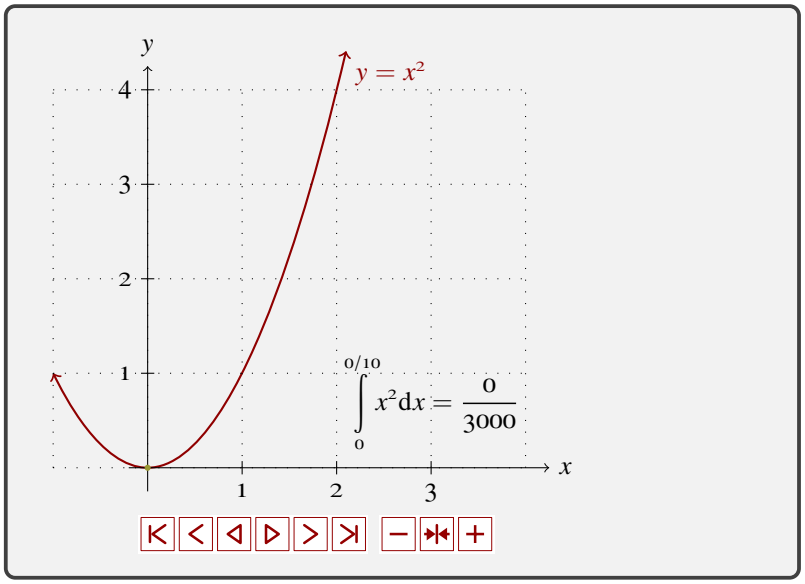
$$\int_a^b f(x) = \lim_{n \to \infty} \sum_{0 \le k \le n} h \cdot f(a + kh)$$

The *integral* is the limit as n goes to ∞ of sums of rectangular areas of height $f(a + kh)$ and width $h = (b - a)/n$.

infinity and the step between samples shrinks to zero.

$$\int_a^b f(x)\,dx = \lim_{h\to 0} \sum_{k=0}^{n} hf(a+kh)$$

where the step between samples is $h = \frac{b-a}{n}$. Note that as $n \to \infty$ the step size h goes to zero.



- The fundamental theorem of calculus states: The integral of a derivative is the difference of function values at the ends of the interval $[a, x]$.

$$\int_a^x f'(t)\,dt = f(x) - f(a)$$

Useful functions f model some behavior that interests us, but these functions may be complex. More simple functions may be able to approximate a complex phenomenon. One simplification is to discretize a continuous function.

For instance,

$$\begin{aligned}
\int_a^b x\,dx &= \lim_{h\to 0} \sum_{k=0}^{n} h(a+kh) \\
&= \lim_{h\to 0} ha(n+1) + h^2 \frac{n(n+1)}{2} \\
&= \lim_{n\to\infty} \frac{b-a}{n} a(n+1) + \frac{(b-a)^2}{n^2} \frac{n(n+1)}{2} \\
&= a(b-a) + \frac{(b-a)^2}{2} \\
&= (b-a)\frac{2a+(b-a)}{2} \\
&= (b-a)\frac{(b+a)}{2} \\
&= \frac{b^2-a^2}{2}
\end{aligned}$$

Discrete Calculus: Sums and Differences

Instantaneous change can be *approximated* by discrete changes. Areas can be *approximated* by sums of discrete areas.

Understanding discrete calculus can help yon understand calculus. Understanding calculus can help you understand discrete calculus. Many problems that can be modeled by calculus can be understood using discrete approximations of continuous models.

The discrete analogs of derivatives and integrals are differences and sums. Discrete numerical mathematics replaces limits of divided differences and infinite sums by more simple ideas.

"Can you do Addition?" the White Queen asked.

"What's one and one and one and one and one and one and one and one and one and one?"

"I don't know," said Alice. "I lost count."

"She can't do Addition," the Red Queen interrupted.

"Can you do Subtraction? Take nine from eight."

Lewis Carroll (Carroll et al., 2000).

Calculus studies functions defined on the real numbers $\mathbb{R}$. Discrete calculus studies functions on sequences of numbers, most often the natural numbers $\mathbb{N}$.

Sample Sequences

There are several famous sequences that can be used to describe the techniques used in the sum & difference calculus. Here is a partial list of useful sequences.

May I recommend the Online Encyclopedia of Integer Sequences to you? On May 4, 2014 it contained 241,515 edited sequences that were found interesting to somebody.

Alice is ... do I have to say?

- Alice: $\vec{A} = \langle 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, \ldots \rangle$

$$\begin{aligned} a(n) &= 1 && \text{functional form} \\ a_n &= a_{n-1},\ a_0 = 1 && \text{recursive-initial value form} \end{aligned}$$

  In HASKELL: `let a = [ 1 | n <- [0..]]`.

  The Alice sequence is fundamental to counting. Using the alphabet $\{1\}$ you can name the natural numbers as *unary* strings

$$1 = 1,\ 2 = 11,\ 3 = 111,\ 4 = 1111,\ 5 = 11111, \ldots$$

  Unary notation wastes space, but its arithmetic is simple.

The well known story of Gauss, in the third grade, summing the natural numbers from 1 to 100 and Gauss' genius at number theory causes me to name the sequence of natural numbers for Gauss (Hayes, 2006).

Numbers in the Gauss sequence obey a linear growth model: Incrementing by a fixed constant (slope) at each step. Steps by 1 are perhaps the most simple example.

- Gauss: $\vec{G} = \langle 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 \ldots \rangle$

$$\begin{aligned} g(n) &= n && \text{functional form} \\ g_n &= g_{n-1} + 1,\ g_0 = 0 && \text{recursive-initial value form} \end{aligned}$$

  In HASKELL: `let g = [ n | n <- [0..]]`.

It seems to me that *connectedness* is the fundamental concept described by *triangular* numbers.

- Triangular: $\vec{T} = \langle 0, 0, 1, 3, 6, 10, 15, 21, 28, 36, 45, \ldots \rangle$

$$\begin{aligned} t(n) &= n(n-1)/2 && \text{functional form} \\ t_n &= t_{n-1} + (n-1),\ t_0 = 0 && \text{recursive-initial value form} \end{aligned}$$

  In HASKELL: `let t = [ n*(n-1)/2 | n <- [0..]]`.

The growth (or death) of many THINGS can be approximated exponentially: Doubling is perhaps the most simple example.

- Doubling (or Power of 2): $\vec{D} = \langle 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, \ldots \rangle$

$$\begin{aligned} d(n) &= 2^n && \text{functional form} \\ d_n &= 2d_{n-1},\ d_0 = 1 && \text{recursive-initial value form} \end{aligned}$$

  In HASKELL: `let d = [ 2^n | n <- [0..]]`.

Mersenne was a "renaissance man" who I would describe as a "core router" on the early 1600s Internet. He is best remembered today because his sequence is where people search for bigger and bigger prime numbers through the GIMPS project.

- Mersenne: $\vec{M} = \langle 0, 1, 3, 7, 15, 31, 63, 127, 255, 511, \ldots \rangle$

$$\begin{aligned} m(n) &= 2^n - 1 && \text{functional form} \\ m_n &= 2m_{n-1} + 1,\ m_0 = 0 && \text{recursive-initial value form} \end{aligned}$$

  In HASKELL: `let m = [ 2^n-1 | n <- [0..]]`.

- Fibonacci: $\vec{F} = \langle 0,\ 1,\ 1,\ 2,\ 3,\ 5,\ 8,\ 13,\ 21,\ 34, \ldots \rangle$

$f(n) = \dfrac{\varphi^n - \bar{\varphi}^n}{\sqrt{5}}$	functional form
$f_n = f_{n-1} + f_{n-2},\ f_0 = 0,\ f_1 = 1$	recursive-initial value form

In HASKELL: `let f = 0 : 1 : (zipWith (+) f (tail f))`.

The HASKELL definition of the Fibonacci sequence is neat. It says,

- The first and second terms of the Fibonacci sequence are 0 and 1.
- The tail of the sequence can be computing by adding the list so far with its tail. For instance, if `f=[0,1,1,2,3,...]` then after the first two terms, the list is given by `f=0:1:[0,1,1,2,3,...]+[1,1,2,3,...]`

The golden ratio is $\varphi = \frac{1+\sqrt{5}}{2} \approx 1.618$. Fibonacci numbers appear to describe the growth of many phenomena: population growth, properties on plants and other living organisms, architectural designs, ...

Quadratic growth is another polynomial growth model.

- Quadratic: $\vec{Q} = \langle 0,\ 1,\ 4,\ 9,\ 16,\ 25,\ 36,\ 49,\ 64,\ 81, \ldots \rangle$

$q(n) = n^2$	functional form
$q_n = q_{n-1} + 2n - 1,\ q_0 = 0$	recursive-initial value form

In HASKELL: `let q = [ n*n | n <- [0..]]`.

Pass a Quiz: Sequence basics

Take a quiz on page 316 to check your understanding. You can return to here from the quiz.

The Difference Sequence

Often terms s_n in a sequence $\vec{S}$ are data obtained by evaluating a function $s(x)$ at integer points $x = 0,\ 1,\ 2, \ldots$. Another sequence source is values from sampling a signal $s(t)$ at discrete times $t = 0,\ 1,\ 2, \ldots$. The change in the signal over time provides information about the sequence.

Definition 15: Difference of a sequence

Let

$$\vec{S} = \langle s_0,\ s_1,\ s_2,\ s_3, \ldots,\ s_n, \ldots \rangle$$

be a sequence. The (forward) difference of $\vec{S}$ is the sequence

$$\triangle\vec{S} = \langle (s_1 - s_0),\ (s_2 - s_1),\ (s_3 - s_2),\ (s_4 - s_3), \ldots,\ (s_{n+1} - s_n), \ldots \rangle$$

where each term in $\triangle\vec{S}$ is the difference of successive terms in $\vec{S}$.

On a term-by-term basis, each term in $\triangle\vec{S}$ is computed by

$$\triangle s_k = s_{k+1} - s_k,\ k = 0,\ 1,\ 2,\ 3, \ldots$$

Here are some results for the sequences mentioned above.

Figure 2: Differences of sample sequences

The difference of the Alice sequence is the *Zero* sequence.

$$\triangle\vec{A} = \langle 0,\ 0,\ 0,\ 0, \ldots \rangle = \vec{Z}$$

The difference of the Gauss sequence is the Alice sequence.

$$\triangle\vec{G} = \langle 1,\ 1,\ 1,\ 1, \ldots\rangle = \vec{A}$$

The difference of the Triangular sequence is the Gauss sequence.

$$\triangle\vec{T} = \langle 0,\ 1,\ 2,\ 3, \ldots\rangle = \vec{G}$$

The difference of the Doubling sequence is itself.

$$\triangle\vec{D} = \langle 1,\ 2,\ 4,\ 8, \ldots\rangle = \vec{D}$$

The difference of the Mersenne sequence is the Doubling sequence.

$$\triangle\vec{M} = \langle 1,\ 2,\ 4,\ 8, \ldots\rangle = \vec{D}$$

The difference of the Fibonacci sequence inserts 1 to the start of the sequence

$$\triangle\vec{F} = \langle 1,\ 0,\ 1,\ 1,\ 2,\ 3,\ 5, \ldots\rangle = 1 : \vec{F}$$

The difference of the Quadratic sequence is the Odd sequence.

$$\triangle\vec{Q} = \langle 1,\ 3,\ 5,\ 7,\ 9,\ 11,\ 13, \ldots\rangle = \vec{O}$$

And, the difference of the Odd sequence is twice the Alice sequence.

$$\triangle\vec{OD}D = \langle 2,\ 2,\ 2,\ 2,\ \ldots\rangle = 2 \cdot \vec{A}$$

In HASKELL, the `diff` function maps a sequence to its difference.

```
diff :: Num n => [n] -> [n]          -- the diff function maps a list to a list
diff xs = zipWith (-) (tail xs) xs} -- subtract, term-by-term,
                                     -- the tail of sequence s from itself.
```

Pass a Quiz: Differences

Take a quiz on page 317 to check your understanding. You can return to here from the quiz.

The Sum Sequence

Let

$$\vec{S} = \langle s_0,\ s_1,\ s_2,\ s_3, \ldots,\ s_n, \ldots\rangle$$

be a sequence. The sequence of $\vec{S}$'s partial sums is the sequence

$$\sum \vec{S} = \left\langle 0,\ s_0,\ s_0 + s_1,\ s_0 + s_1 + s_2,\ \ldots,\ \sum_{k=0}^{n-1} s_k, \ldots \right\rangle$$

I choose to always begin a sequence of partial sums with the empty sum $\sum_0^{-1}$ (the sum of no terms) which is equal to zero. These partial sums are also called prefix sums.

This choice seems to make sense in all the instances I've considered, and it make for simple initial or terminal conditions.

It is useful to name a sum, for example, you could use $S(n)$ to name the sum of n terms in a sequence $\vec{S}$.

$$S(n) = \sum_{k=0}^{n-1} s_k = s_0 + s_1 + s_2 + \cdots + s_{n-1}$$

On a term-by-term basis, each term in $\sum \vec{S}$ is computed by

$$\sum_{k=0}^{n-1} s_k, \quad n = 0,\ 1,\ 2,\ 3, \ldots$$

That is, let n be the number of terms in a sum. Then the first few partial sums of terms in $\vec{S}$ are:

$n = 0$	$\sum_{k=0}^{-1} s_k = 0$	the sum of no terms
$n = 1$	$\sum_{k=0}^{0} s_k = s_0$	the sum of 1 term
$n = 2$	$\sum_{k=0}^{1} s_k = s_0 + s_1$	the sum of 2 terms
$n = 3$	$\sum_{k=0}^{2} s_k = s_0 + s_1 + s_2$	the sum of 3 terms

Here are some examples for the sequences mentioned above.

Figure 3: Sums of sample sequences

The sum of the Alice sequence is the Gauss sequence.

$$\sum \vec{A} = \langle 0,\ 1,\ 2,\ 3, \ldots \rangle = \vec{G}$$

The sum of the Gauss sequence is the Triangular sequence.

$$\sum \vec{G} = \langle 0,\ 0,\ 1,\ 3,\ 6,\ 10,\ 15, \ldots \rangle = \vec{T}$$

The sum of the Triangular sequence is the *Pyramidal* sequence.

$$\sum \vec{T} = \langle 0,\ 0,\ 0,\ 1,\ 4,\ 10,\ 20, \ldots \rangle = \vec{P}$$

The sum of the Doubling sequence is the Mersenne sequence.

$$\sum \vec{D} = \langle 0,\ 1,\ 3,\ 7,\ 15, \ldots \rangle = \vec{M}$$

The sum of the Mersenne sequence is the Doubling minus the tail of the Gauss sequence.

$$\sum \vec{M} = \langle 0,\ 0,\ 1,\ 4,\ 11,\ 26, \ldots \rangle = \vec{D} - \textsc{tail}(\vec{G})$$

The sum of the Fibonacci sequence is the tail of itself minus the Alice sequence.

$$\sum \vec{F} = \langle 0,\ 0,\ 1,\ 2,\ 4,\ 7,\ 12, \ldots \rangle = \textsc{tail}(\vec{F}) - \vec{A}$$

The sum of the Quadratic sequence is not simple.

$$\sum \vec{Q} = \langle 0,\ 0,\ 1,\ 5,\ 14,\ 30,\ 55, \ldots \rangle$$

And, the sum of the Odd sequence is the Quadratic sequence.

$$\sum \vec{ODD} = \langle 0,\ 1,\ 4,\ 9,\ 16,\ \ldots \rangle = \vec{Q}$$

In HASKELL, the `sumSeq` function maps a sequence to its partial sums.

```
sumSeq :: Num n => [n] -> [n]  -- the function maps a list to a list
sumSeq [] = [0]                -- the of terms in an empty list is 0
sumSeq ns = scanl (+) 0 ns     -- add terms in ns from the left,
                               -- starting with 0
```

In the sum

$$\sum_{k=0}^{n-1} s_k,\ n = 0,\ 1,\ 2,\ 3, \ldots$$

$k = 0$ is called the lower limit, $k = n - 1$ is the upper limit, and s_k are the terms being added.

Pass a Quiz: Sums

Take a quiz on page 317 to check your understanding. You can return to here from the quiz.

The Fundamental Theorem of the Sum & Difference Calculus

The Fundamental theorem of discrete calculus states that the sum of differences telescopes:

$$(s_1 - s_0) + (s_2 - s_1) + \cdots + (s_n - s_{n-1}) = s_n - s_0$$

Note the analogy between formulas:

$$\int_a^b \frac{d}{dx} f(x)\, dx = f(x)\Big|_a^b = f(b) - f(a)$$

and

$$\frac{d}{dx} \int_a^x f(x)\, dx = f(x)$$

Theorem 10: Fundamental theorem of discrete (sum & difference) calculus

A sum of differences is

$$\sum_{k=0}^{n-1} \triangle s_k = s_k\Big|_0^n = s_n - s_0$$

Furthermore, the difference of a sum is

$$\triangle\left(\sum_{k=0}^{n-1} s_k\right) = \left(\sum_{k=0}^{n} s_k - \sum_{k=0}^{n-1} s_k\right) = s_n$$

Proof: Fundamental theorem of discrete calculus

This fundamental theorem is easy to prove: A sum of difference telescopes.

$$\begin{aligned}\sum_{k=0}^{n-1} \triangle s_k &= \sum_{k=0}^{n-1} (s_{k+1} - s_k) \\ &= (s_1 - s_0) + (s_2 - s_1) + (s_3 - s_2) + \cdots + (s_n - s_{n-1}) \\ &= s_k \Big|_0^n \\ &= s_n - s_0\end{aligned}$$

Furthermore,

$$\begin{aligned}\triangle\left(\sum_{k=0}^{n-1} s_k\right) &= (s_0 + s_1 + s_2 + \cdots + s_n) - (s_0 + s_1 + s_2 + \cdots + s_{n-1}) \\ &= s_n\end{aligned}$$

Pass a Quiz: Fundamental Theorem

Take a quiz on page 318 to check your understanding. You can return to here from the quiz.

Homework Questions

Use your time outside of class to solve these problems.

1 Consider the sequence of even natural numbers

$$\mathbb{EVEN} = \{0, 2, 4, 6, 8, \ldots\}$$

Write the functional and recursive-initial value form that describe elements in $\mathbb{EVEN}$.

2 Consider the sequence of odd natural numbers

$$\mathbb{ODD} = \{1, 3, 5, 7, 9, \ldots\}$$

Write the functional and recursive-initial value form that describe elements in $\mathbb{ODD}$.

3 Verify the differences of terms in the Alice, Gauss, Triangular, Doubling, Mersenne, Fibonacci, and Quadratic sequences are as claimed in Figure 2.

4 Use the fundamental theorem of discrete calculus to verify the partial sums of terms in the Alice, Gauss, Triangular, Doubling, Mersenne, Fibonacci, and Quadratic sequences are as claimed in Figure 3.

5 Most people define the triangular numbers by the function

$$t(n) = \frac{n(n+1)}{2} \quad \text{for } n \in \mathbb{N}$$

What is the difference, if any, between this definition and the one used in these notes?

6 Consider the sequence of *triangular fractions*

$$\left\langle 1, \frac{1}{3}, \frac{1}{6}, \frac{1}{10}, \frac{1}{15}, \ldots \right\rangle$$

6.1 What function computes terms in this sequence?

6.2 Use the idea that difference of Triangular numbers are Gauss numbers to show that terms in the *triangular fraction* sequence can be written as differences of *Gauss fractions*.

$$\frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \cdots + \frac{1}{(n-1)) \cdot n} + \frac{1}{n \cdot (n+1)}$$

6.3 Show that each term $\frac{1}{k \cdot (k+1)}$ in the sum can be written as the difference

$$\frac{1}{k \cdot (k+1)} = \frac{1}{k} - \frac{1}{k+1} = -\Delta \frac{1}{k}$$

6.4 Use the fundamental theorem of the sum & difference calculus to find a formula for the sum.

7 Consider the sum

$$\frac{1}{2!} + \frac{2}{3!} + \cdots + \frac{n}{(n+1)!} + \frac{n+1}{(n+2)!}$$

7.1 Show that each term $\frac{k}{(k+1)!}$ in the sum can be written as the difference

$$\frac{k}{(k+1)!} = \frac{1}{k!} - \frac{1}{(k+1)!}$$

7.2 Use the fundamental theorem of the sum & difference calculus to find a formula for the sum.

8 Consider the sequence of ratios of Fibonacci numbers

$$\left\langle \frac{F_2}{F_1}, \frac{F_3}{F_2}, \ldots, \frac{F_{(n+1)}}{F_n}, \ldots \right\rangle$$

8.1 Write the first seven ratios

$$\left\langle \frac{1}{1}, \frac{2}{1}, \frac{3}{2}, \frac{5}{3}, \frac{8}{5}, \frac{13}{8}, \frac{21}{13} \right\rangle$$

in fixed-point decimal notation. Guess the value to which these ratios converge.

8.2 In calculus, you'll learn about limits and be required to show results such as *the limit of ratios of Fibonacci numbers is the golden ratio* φ.

$$\lim_{n\to\infty} \frac{F_{(n+1)}}{F_n} = \varphi$$

Use your knowledge that Fibonacci numbers can be computed by the function

$$f(n) = \frac{\varphi^n - \bar{\varphi}^n}{\sqrt{5}}$$

to show the above statement is `True`.

8.3 1 mile is 1.60934 kilometers, or 1 kilometer is 0.621371 miles. The golden ratio $\varphi \approx 1.618033$ and $1/\varphi \approx 0.61803$ are reasonable approximations to these conversion factors.

8.3.1 What is the relative error in both cases?

8.3.2 Use this approximation to conclude F_n kilometers is approximately F_{n-1} miles.

9 Consider the sequence of natural numbers

$$0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9,\ 10,\ 11,\ 12,\ 13,\ 14,\ 15,\ 16, \ldots$$

9.1 Circle every other number and sum the uncircled one. What is a formula for the sum?

9.2 Circle every third number and sum the uncircled one. What is a formula for the sum?

Pass a Quiz: Summative exam #2 on functions, Horner's rule, and sequences

Take a quiz on page 319 to check your understanding. You can return to here from the quiz.

8. Machine numbers: How computers do arithmetic

. . . if a program or algorithm looks pretty, why then, it must be OK. If you think that beauty is the sole criterion, remember that beauty is in the eye of the beholder, and in the eyes of a bug, a rose is just fodder!

You have to learn the techniques of error analysis and decide what degree of error is tolerable. There are three good books: "Accuracy and Stability of Numerical Algorithms," by Nicholas J. Higham (Higham, 1996), "Matrix Computations," Third Edition, by Gene H. Golub and Charles F. Van Loan (Golub and Van Loan, 1996), and "The Algebraic Eigenvalue Problem," by J. H. Wilkinson (Wilkinson, 1988).

A Conversation with William Kahan, Dr. Dobb's Journal, 271, pp. 18-32, 1997.

Computers and calculators do not always follow the rules of mathematics. Ultimately, computer numbers are bounded and finite, while mathematical numbers are unbounded: Real numbers can become arbitrarily close to 0 and can tend to infinity (∞) without bound.

Sometimes 3-digit decimal examples are used.

Machine numbers will be explained in terms of 8-bit computer words as describe in the notes on abstracting a computer. Some fundamental ideas were described there and in the notes on arithmetic. You may want to refresh your memory. You will be expected to generalize from 8-bit examples to 16, 32, or 64 bit examples that model modern electronic computers. There are three number *types* we will consider:

1. Unsigned natural numbers $\mathbb{N} = \{0,\ 1,\ 2,\ 3, \ldots\}$
2. Signed integers $\mathbb{Z} = \{0,\ \pm 1,\ \pm 2,\ \pm 3, \ldots\}$
3. Rational floating-point numbers $\mathbb{Q}_{fp} = \{\pm 1.f \times 2^e : f \in \mathbb{B}^4,\ e \in \mathbb{Z}_{\text{bias}=3}\}$ where $p = 4$ is the *precision* and $b = 3$ is the *exponent bias*.

$\mathbb{B}^p$ is the set of all p bit strings: There are 2^p such strings. The set $\mathbb{Z}_{\text{bias}=b}$ is the set of biased integers: A value $n \in \mathbb{Z}_{\text{bias}=b}$ represents the integer $n - b$. More details will be provided below.

Unsigned natural numbers

On an 8-bit computer, $2^8 = 256$ unsigned natural numbers can be *named*: 0, 1, . . . , up to 255. The smallest unsigned natural number is 0, stored in an 8-bit *word*.

$$0 = \{ \begin{array}{cccccccc} 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ \hline \boxed{0} & \boxed{0} & \boxed{0} & \boxed{0} & \boxed{0} & \boxed{0} & \boxed{0} & \boxed{0} \end{array}$$

The largest unsigned integer is $2^8 - 1 = 255 = (FF)_{16}$, stored in an 8-bit *word*.

$$255 = \{ \begin{array}{cccccccc} 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

Other natural numbers between 0 and 255 are stored using different bit patterns. For instance,

$$73 = \{ \begin{array}{cccccccc} 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ \hline 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{array} \qquad 237 = \{ \begin{array}{cccccccc} 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ \hline 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{array}$$

Addition on 8-bit numbers is performed modulo 256.

When 1 is added to 255 our computer will compute 0 as the sum. Our computer can detect this error because the carry-bit will be 1 and the zero-bit will be 0. Other strange arithmetic can happen too, for instance

Do you see why $73 + 237 = 54$? If not, ask someone who knows.

$73+237 = 54$ while 310 is what third grade math would tell us it should be.

Signed numbers (integers)

In third grade mathematics negative integers are written in sign-magnitude notation. An explicit minus sign $(-)$ is used to denote negative integers. The plus sign is usually dropped for positive integers, but computers need an explicit sign to distinguish positive from negative integers.

Sign-magnitude notations leads to two different zeros. The logic for sign-magnitude arithmetic is harder to implement than other systems. This "difficult" system may be one of the reasons children face problems when learning to add, subtract, multiply and divide.

$$-41, -9, -1, -0, +0, +1, +9, +41$$

Instead of sign-magnitude notation, computers represent signed integers using complement notation. The basic idea is that an integer and its negative sum to zero. For instance, using a decimal computer with words of length 3 you can write strings from 000 to 999. Consider the sums:

$$\begin{array}{r} 324 \\ \underline{676} \\ 1000 \end{array} \qquad \begin{array}{r} 563 \\ \underline{437} \\ 1000 \end{array} \qquad \begin{array}{r} 242 \\ \underline{758} \\ 1000 \end{array} \qquad \begin{array}{r} 777 \\ \underline{223} \\ 1000 \end{array} \qquad \begin{array}{r} 151 \\ \underline{849} \\ 1000 \end{array}$$

Each sum is equal to 000 when the overflow into a fourth digit is dropped. That is, these pairs of integers are 3-digit ten's complement *negatives* of each other. The interpretation is:

- Names $(n)_{10c}$ between $(000)_{10c}$ and $(499)_{10c}$ represent their normal non-negative values

  $$(0)_{10c} = 0 \qquad (324)_{10c} = 324 \qquad (223)_{10c} = 223 \qquad (499)_{10c} = 499$$

- Names $(n)_{10c}$ between $(500)_{10c}$ and $(999)_{10c}$, are 1000 more than the negative numbers they represent.

  $$(500)_{10c} = -500 \qquad (676)_{10c} = -324 \qquad (777)_{10c} = -223 \qquad (999)_{10c} = -1$$

In general, three digit string from 000 to 999 are interpreted to be integers in the following way.

Three-digit ten's complement numbers

$$(x)_{10c} = \begin{cases} x & \text{if } 000 \le x \le 499 \\ x - 1000 & \text{if } 500 \le x \le 999 \end{cases}$$

In this way

$$(x)_{10c} + (1000 - x)_{10c} = (1000)_{10c} = 0$$

For instance, with $x = 237$, we have

$$(237)_{10c} + (1000 - 237)_{10c} = (237)_{10c} + (763)_{10c} = 237 + (763 - 1000) = 0$$

Padding ten's complement numbers

Ten's complement numbers can be padded to fill larger registers. For instance, to expand a 3-digit computer to 6-digits:

- Pad positive numbers with 0's on the left.

  $(073)_{10c} = (000\ 073)_{10c} \qquad (123)_{10c} = (000\ 123)_{10c} \qquad (499)_{10c} = (000\ 499)_{10c}$

- Negative numbers are padded with 9's on the left. For instance,

  $(856)_{10c} = (999\ 856)_{10c} \qquad (737)_{10c} = (999\ 737)_{10c} \qquad (500)_{10c} = (999\ 500)_{10c}$

Pass a Quiz: Machine numbers basics

Take a quiz on page 321 to check your understanding. You can return to here from the quiz.

Two's complement numbers

Two's complement numbers are written using the binary alphabet $\mathbb{B} = \{0, 1\}$. This is the number system used for integer arithmetic on modern electronic computers. For illustration of two's complement ideas we'll use 8-bit binary strings.

$$(b_7 b_6 b_5 b_4\ b_3 b_2 b_1 b_0)_{2c}$$

These 256 strings will represent integers from -128 to 127.

In general, 8-bit strings from 0000 0000 to 1111 1111 are interpreted as integers in the following way.

Eight-bit two's complement numbers

$$(x)_{2c} = \begin{cases} x & \text{if } 0 \le x \le 127 \\ x - 256 & \text{if } 128 \le x \le 255 \end{cases}$$

The leading leftmost bit b_7 signals if the number is positive or negative. But it is not just a sign. It is more than that. It is best to think of the eighth bit b_7 to represent the value $(-1)^{b_7}$.

Bit b_7 changes from 0 to 1 as you count from 127 to 128.

$w = (1010\ 0000)_{2c} = -96$ is a negative number; its negative is the positive number $-w = (0110\ 0000)_{2c} = 96$.

$$b_7 = \begin{cases} 0 & \text{the number is positive (or perhaps 0).} \\ 1 & \text{the number is negative.} \end{cases}$$

In this context, the bit-pattern stored in a word is decoded differently from unsigned natural numbers.

	7	6	5	4	3	2	1	0
0 = {	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
-1 = {	1	1	1	1	1	1	1	1

	7	6	5	4	3	2	1	0
73 = {	0	1	0	0	1	0	0	1

	7	6	5	4	3	2	1	0
-19 = {	1	1	1	0	1	1	0	1

Complement notation is defined by the rule that a number and its negative sum to 0. For instance, consider the 8-bit numbers $w = (1001\ 1100)_{2c}$, whose negative is $-w = (0110\ 0100)_{2c}$. That's because their sum is:

Be careful of words. $w = (1001\ 1100)_{2c} = -100$ is a negative number, and its negative is $-w = (0110\ 0100)_{2c} = 100$ is a positive number.

Carries	1	1	1	1	1	1	0	0	0
$w =$		1	0	0	1	1	1	0	0
$-w =$		0	1	1	0	0	1	0	0
Sum	1	0	0	0	0	0	0	0	0

The sum is eight 0 bits with an overflow carry bit of 1. Our 8-bit computer records 0000 0000, that is 0.

Our example two's complement numbers are 8-bits wide: $2^8 = 256$ different values can be represented.

- The most negative 8-bit two's complement number is

$$(1000\ 0000)_{2c} = -128$$

- The most positive 8-bit two's complement number is

$$(0111\ 1111)_{2c} = +127$$

There are 256 integers from -128 to 127, inclusive.

Suppose instead, computer words are 4-bits wide. The table below show the natural numbers for 0 to 15 (0 to F, 0000 to 1111) and how these bit patterns represent integers in two's complement and biased notations.

Natural numbers	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Two's Complement	0	1	2	3	4	5	6	7	-8	-7	-6	-5	-4	-3	-2	-1
Biased $b = 7$	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8

Negating a two's complement number: Copy and flip

A two's complement number can be negated (positive-to-negative or negative-to-positive) using a simple bit-flipping rule.

1. Copy the bits from right-to-left up to and including the first 1. For instance,

$$(0111\ 1000)_{2c} \overset{\text{copy}}{\mapsto} (\cdots\ 1000)_{2c} \quad \text{or} \quad (1000\ 1111)_{2c} \overset{\text{copy}}{\mapsto} (\cdots 1)_{2c}$$

2. Flip ($0 \mapsto 1$, $1 \mapsto 0$) the remaining bits to the left For instance,

$$(0111\ 1000)_{2c} \overset{\text{flip}}{\mapsto} (1000\ 1000)_{2c} \quad \text{or} \quad (1000\ 1111)_{2c} \overset{\text{copy}}{\mapsto} (0111\ 0001)_{2c}$$

For instance, the negative of

$$(1000\ 1000)_{2c} \qquad \text{is} \qquad \begin{pmatrix} \overset{\text{flip}}{0111} & \overset{\text{copy}}{1000} \end{pmatrix}_{2c}.$$

Notice that the negative of -128 cannot be represented using 8 bits. The largest 8-bit two's complement number is 127. Nine bits are required to represent $+128$.

$$(0\ 1000\ 0000)_{2c} = +128$$

The bit-flipping algorithm to negate a two's complement integer is not the hardware implementation. Most authors will describe two's complement negation by

1. Compute the one's complement: Flip each bit in the binary string.
2. Add 1 to the one's complement.

Convince yourself that these steps produce the same result as described in this course.

Using the bit-flipping rule to negate an integer in two's complement notation goes like this:

				←				
			Flip here				Copy here	
-76	1	0	1	1	0	1	0	0
↓	↓	↓	↓	↓	↓	↓	↓	↓
76	0	1	0	0	1	1	0	0

The bit-flipping rule fails for a string that starts with 1 followed by only 0's. Well the input is the output in this case. The most negative two's complement number in n-bits can only be represented in $n+1$ bits.

Converting from two's complement to decimal

First, you can convert a positive two's complement number to its decimal equivalent by using Horner's rule. If the number is positive (its leading, most significant bit is 0), simply use Horner's rule. For instance, $(0010\ 1100)_{2c} = 44$.

Or you can do the conversion in your head if the number is small enough. Using look-up table can be a good method for small problems.

Horner's Rule @ $x = 2$

0	0	1	0	1	1	0	0
	0	0	2	4	10	22	44
0	0	1	2	5	11	22	$\boxed{44}$

Negative two's complement numbers can be converted to their decimal form in several ways. Here are three.

1. I think students prefer this way:

   Pretend the two's complement number is negative, its leading, leftmost bit is 1. Then, use the bit-flipping rule to negate it; apply Horner's rule; and negate the result, for instance, you can show $(1101\ 0100)_{2c} = -44$.

$$(1101\ 0100)_{2c} \overset{\text{negate}}{\mapsto} (0010\ 1100)_{2c} \overset{\text{Horner}}{\mapsto} 44 \overset{\text{negate}}{\mapsto} -44$$

   Here's another example. To convert $(1100\ 0100)_{2c}$ to it's decimal equivalent

- Negate it.

$$(1100\ 0100)_{2c} \overset{\text{negate}}{\mapsto} (0011\ 1100)_{2c}$$

- Evaluate the negative using Horner's rule.

Horner's Rule @ $x = 2$

0	0	1	1	1	1	0	0
	0	0	2	6	14	30	60
0	0	1	3	7	15	30	$\boxed{60}$

- Negate it. $-60 = (1100\ 0100)_{2c}$

And here's another example that shows

$$(1010\ 1101)_{2c} = -83$$

First, negate the two's complement number

$$-(1010\ 1101)_{2c} = (0101\ 0011)_{2c}$$

Next, evaluate the negation using Horner's rule.

Horner's Rule @ $x = 2$

0	1	0	1	0	0	1	1
	0	2	4	10	20	40	82
0	1	2	5	10	20	41	$\boxed{83}$

Now negate the result to get $(1010\ 1101)_{2c} = -83$.

2. Here's a second way to convert a negative two's complement number to its decimal equivalent.

   Use Horner's rule on the two's complement number and then subtract 256 from the result. For instance, to convert $(1010\ 1101)_{2c}$ to decimal, compute

This method depends on the word length. Using 8-bit words, subtract $2^8 = 256$ from the output of Horner's rule. When the two's complement number is 9-bits long, subtract $2^9 = 512$. For a 32-bit word, subtract 2^{32}.

Horner's Rule @ $x = 2$

1	0	1	0	1	1	0	1
	2	4	10	20	42	86	172
1	2	5	10	21	43	86	$\boxed{173}$

Now subtract $2^8 = 256$ from 173.

$$(1010\ 1101)_{2c} = 173 - 2^8 = 173 - 256 = -83$$

Here is another example showing $(1100\ 0100)_{2c} = -60$

Horner's Rule @ $x = 2$

1	1	0	0	0	1	0	0
	2	6	12	24	48	98	196
1	3	6	12	24	49	98	$\boxed{196}$

Subtract 256 to determine

$$(1100\ 0100)_{2c} = 196 - 256 = -60$$

I just learned of third conversion from a comment made by Roger B.

3. Finally, here's a third way to convert a negative two's complement number to its decimal equivalent.

   Treat the leading, leftmost bit $b_7 = 1$ as $(-1)^{b_7} = -1$ and use Horner's rule to the resulting string. That is, to convert

$$(1b_6b_5b_4\ b_3b_2b_1b_0)_{2c}$$

to decimal by applying Horner's rule to the string

$$-1b_6b_5b_4\ b_3b_2b_1b_0$$

For example, to compute the decimal equivalent of $(1010\ 1101)_{2c}$, apply Horner's rule to $-1, 0, 1, 0, 1, 1, 0, 1$

Horner's Rule @ $x = 2$

-1	0	1	0	1	1	0	1
	-2	-4	-6	-12	-22	-42	-84
-1	-2	-3	-6	-11	-21	-42	$\boxed{-83}$

Here is another example showing $(1100\ 0100)_{2c} = -60$

Horner's Rule @ $x = 2$

-1	1	0	0	0	1	0	0
	-2	-2	-4	-8	-16	-30	-60
-1	-1	-2	-4	-8	-15	-30	$\boxed{-60}$

And here's another example.

Horner's Rule @ $x = 2$

-1	1	0	0	1	0	0	0
	-2	-2	-4	-8	-14	-28	-56
-1	-1	-2	-4	-7	-14	-28	$\boxed{-56}$

That is, $(1100\ 1000)_{2c} = -56$

You should question why this works. Consider: We are using 8 bits, two's complement integers to represent positive and negative integers.

Let $(b_7b_6b_5b_4\ b_3b_2b_1b_0)_2$ be the binary representation of a natural number $0 \le n \le 255$. The meaning of the two's complement number $(b_7b_6b_5b_4\ b_3b_2b_1b_0)_{2c}$ is given by the two cases for the most significant bit b_7.

$$(b_7b_6b_5b_4\ b_3b_2b_1b_0)_{2c} = \begin{cases} (0b_6b_5b_4\ b_3b_2b_1b_0)_2 & \text{if } 0 \le n \le 127, \quad b_7 = 0 \\ (1b_6b_5b_4\ b_3b_2b_1b_0)_2 - 256 & \text{if } 128 \le n \le 255, \quad b_7 = 1 \end{cases}$$

Notice the leading 1 in $(1b_6b_5b_4\ b_3b_2b_1b_0)_2$ stands for $2^7 = 128$. Therefore,

$$\begin{aligned} (1b_6b_5b_4\ b_3b_2b_1b_0)_2 - 256 &= (128 + (b_6b_5b_4\ b_3b_2b_1b_0)_2) - 256 \\ &= -128 + (b_6b_5b_4\ b_3b_2b_1b_0)_2 \end{aligned}$$

Therefore, $(1b_6b_5b_4\ b_3b_2b_1b_0)_2$ can be convert to decimal by applying Horner's rule to $(-1b_6b_5b_4\ b_3b_2b_1b_0)_2$. Here's the 4-bit examples.

4-bit negative two's complement numbers

-8	$=$	$(1000)_{2c}$	$=$	$-2^3 + 0$
-7	$=$	$(1001)_{2c}$	$=$	$-2^3 + 1$
-6	$=$	$(1010)_{2c}$	$=$	$-2^3 + 2$
-5	$=$	$(1011)_{2c}$	$=$	$-2^3 + 3$
-4	$=$	$(1100)_{2c}$	$=$	$-2^3 + 4$
-3	$=$	$(1101)_{2c}$	$=$	$-2^3 + 5$
-2	$=$	$(1110)_{2c}$	$=$	$-2^3 + 6$
-1	$=$	$(1111)_{2c}$	$=$	$-2^3 + 7$

In HASKELL, the code to convert from a binary string to decimal might look like this.

```
twoscompToDecimal :: [Binary] -> Integer
twoscompToDecimal t = if head t == 0
     then horner tail t
     else horner tail t - 2**(length t)
```

Converting from decimal to two's complement

Now consider the problem of converting from decimal to two's complement. Again, break the problem down into two cases: positive and negative.

For a positive integer n do the following:

1. Convert n to an *unsigned* binary by repeated remainder, as described in the chapter on Horner's rule.

2. Prepend (on the left) a leading 0 to indicate $+n$.

Don't be someone who fails to use a sign for all signed numbers.

For instance, to write $+73$ in two's complement notation: Use repeated remainder to convert 73 into binary.

Repeated Remaindering mod 2

QUOTIENTS	73	36	18	9	4	2	1
REMAINDERS	1	0	0	1	0	0	1

Therefore, *unsigned* 73 is $(1001001)_2$. Next, prepend a sign to get the two's complement representation of $+73$.

$$+73 = (01001001)_{2c}$$

There are several methods that convert a negative integer n into two's complement notation. Compare with the three ways of changing two's complement to decimal notation.

1. You can negate a negative number, perform repeated remaindering, pad with a sign, and negate the result.

   For instance, to write -37 in two's complement notation: First use repeated remainder to convert $|-37| = 37$ into binary.

   Repeated Remaindering mod 2

QUOTIENTS	37	18	9	4	2	1
REMAINDERS	1	0	1	0	0	1

   Therefore unsigned 37 can be written as $(10\ 0101)_2$ in binary. Next, prepend a 0 to form signed $+37$.

   $$+37 = (010\ 0101)_{2c}$$

   And then use the bit-flipping rule to negate $(010\ 0101)_{2c}$.

   $$-37 = (101\ 1011)_{2c} \quad \text{in 8-bits} \quad -37 = (1101\ 1011)_{2c}$$

   As another example, consider mapping -237 to its two's complement representation. First, by remaindering,

   Repeated Remaindering mod 2

QUOTIENTS	237	118	59	29	14	7	3	1
REMAINDERS	1	0	1	1	0	1	1	1

   unsigned $237 = (1110\ 1101)_2$, signed $+237 = (0\ 1110\ 1101)_{2c}$, and $-237 = (1\ 0001\ 0011)_{2c}$.

2. Using 8-bit numbers, the second method is to add 256 to the negative number and then used repeated remaindering. For instance, to write -37 in two's complement notation, first add 256 to get $-37 + 256 = 219$ and then compute remainders.

For this second method the value to add depends on the magnitude of n. Consider the examples:

$$-2 \le n < -1 \text{add } 4$$
$$-4 \le n < -2 \text{add } 8$$
$$-128 \le n < -64 \text{add } 256$$
$$-2^m \le n < -2^{m-1} \text{add } 2^{m+1}$$

Repeated Remaindering mod 2

QUOTIENTS	219	109	54	27	13	6	3	1
REMAINDERS	1	1	0	1	1	0	1	1

Therefore,

$-37 = (1101\ 1011)_{2c}$ which can be truncated to 7 bits $-37 = (101\ 1011)_{2c}$

3. A third method to convert a negative decimal integer n to two's complement is to simply run the remaindering algorithm on n, being careful when computing quotients.

Note that

$$-37 = 2(-19) + 1$$
$$-19 = 2(-10) + 1$$
$$-10 = 2(-5) + 0$$
$$-5 = 2(-3) + 1$$
$$-3 = 2(-2) + 1$$
$$-2 = 2(-1) + 0$$

Repeated Remaindering mod 2

QUOTIENTS	-37	-19	-10	-5	-3	-2	-1
REMAINDERS	1	1	0	1	1	0	1

Therefore,

$$-37 = (101\ 1011)_{2c}$$

Here is another example.

Repeated Remaindering mod 2

QUOTIENTS	-237	-119	-60	-30	-15	-8	-4	-2	-1
REMAINDERS	1	1	0	0	1	0	0	0	1

Therefore,

$$-237 = (1\ 0001\ 0011)_{2c}$$

Padding two's complement numbers

Two's complement numbers can be padded to fill larger registers. For instance, to expand a 8-bit computer to 16-bits:

- Pad positive numbers with 0's on the left, most significant end.

$(0010\ 1110)_{2c} = (0000\ 0000\ 0010\ 1110)_{2c}$ $(0111\ 0001)_{2c} = (0000\ 0000\ 0111\ 0001)_{2c}$

- Negative numbers are padded with 1's on the left, most significant end. For instance,

$$(1000\ 1000)_{2c} = (1111\ 1111\ 1000\ 1000)_{2c} \qquad (1110\ 1110)_{2c} = (1111\ 1111\ 1110\ 1110)_{2c}$$

Pass a Quiz: Two's complement notation

Take a quiz on page 321 to check your understanding. You can return to here from the quiz.

Explain how two's complement comparison is accomplished.

Floating-point numbers

The set of floating-point numbers is used to approximate the set of real numbers. It is beyond the scope of this course to provide a complete description of floating-point numbers. The IEEE Standard for Floating Point Arithmetic (IEEE 754) describes floating-point numbers in depth. There are some essential ideas you should know before writing answers to problems that involve real numbers.

Floating-point notation is built upon scientific notation. Rational numbers can be written in normalized decimal scientific notation:

$$t = \pm d.f \times 10^e$$

where d is a non-zero digit, f is a decimal string, and e is an integer.

For instance, the *normalized* scientific notation for 0.577215 is 5.77215×10^{-1}.

Restricting to one non-zero digit to the left of the point is called normalizing the floating-point number. This avoids multiple representations for identical values. For instance, $-0.0060221415 \times 10^{26}$ is normalized as $-6.0221415 \times 10^{23}$.

Rational numbers can also be written in *normalized binary* scientific notation:

$$t = \pm 1.f \times 2^e$$

It is not necessary to record the 1 in $1.f$, because it will always be there for *normalized* floating-point numbers. You only need to record the sign $\pm$, the fractional part f, and the exponent e.

7	6 5 4	3 2 1 0
±	Exp	Fraction

I call the examples "pidgin" floating-point numbers. A floating-point number $t \in \mathbb{Q}_{fp}$ is an 8-bit string that is parsed into three parts as shown below.

Storage of 8-bit pidgin floating point numbers

Sign Bit	Exponent (bias = 3)	Fraction
s	$e_2e_1e_0$	$f_{-1}f_{-2}f_{-3}f_{-4}$

The sign bit s controls whether the floating-point number is positive or negative. The next three bits $e = e_2e_1e_0$ encode a base 2 exponent written with a bias $b = 3$. That means the biased exponents in the range $\langle 000, \ldots, 111 \rangle$ represent the integers in the range $\langle -3, \ldots, 4 \rangle$.

The sign bit s conveniently encodes the sign of t by the function

$$(-1)^s = \begin{cases} 1 & \text{when } s = 0 \\ -1 & \text{when } s = 1 \end{cases}$$

Suppose the bias is $b = 3$. The chart below shows biased numbers and their equivalent integers.

Biased Number	Integer
$(0)_{\text{bias}=3}$	-3
$(2)_{\text{bias}=3}$	-1
$(3)_{\text{bias}=3}$	0
$(4)_{\text{bias}=3}$	1
$(7)_{\text{bias}=3}$	4

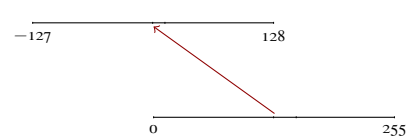
Suppose the bias is $b = 8$. The chart below shows some biased numbers and their equivalent integers.

Biased Number	Integer
$(0)_{\text{bias}=8}$	8
$(8)_{\text{bias}=8}$	0
$(15)_{\text{bias}=8}$	7

Suppose the bias $b = 127$. Then the biased representation of $n = -37$ is

$$(-37 + 127)_{\text{bias}=127} = (90)_{\text{bias}=127}$$

The unsigned integers 0 to 255 can be translated left by $b = 2^7 - 1 = 127$ mapping them from -127 to 127.



Biased notation

Biased notation is an alternative to signed-magnitude and two's complement notation for representing integers. Integer n is represented in biased notation as

$$(n)_{\text{bias}=b} = n - b, \quad \text{where the } \textit{bias } b > 0.$$

Biased notation is a simple left-linear shift of all integers n by an amount b. Adding a bias $b > 0$ to an integer n *translates* each integer n to the right by b places.

In the pidgin floating-point notation, biased numbers lie between 0 and $7 = 2^3 - 1$.

$$0 \leq (n)_{\text{bias}=4} \leq 7$$

With a bias is $b = 3$. This gives integer exponents n in the range

$$-3 \leq n \leq 4$$

where

$$-3 = 000, , \ -2 = 001, \ -1 = 010, \ 0 = 011, \ 1 = 100, \ 2 = 101, \ 3 = 110, \ 4 = 111$$

In practice, biased numbers lie between 1 and $2^m - 2$, for some number of bits m.

$$1 \leq (n)_{\text{bias}=b} \leq 2^m - 2$$

And, the bias is $b = 2^{m-1} - 1$.

For IEEE 754 numbers the biases are:

- Single-precision floating point numbers are 32 bits wide. Their exponents are 8 bits wide and written as biased numbers in the range 1 to $254 = 2^8 - 2$. Their bias is $b = 127 = 2^7 - 1$. This gives a range of exponents from

$$-126 = 1 - 127 \quad \text{to} \quad 127 = 254 - 127$$

- Double-precision floating point numbers are 64 bits wide. Their exponent are 11 bits, written as biased numbers in the range 1 to $2046 = 2^{11} - 2$. The bias is $b = 1023 = 2^{10} - 1$. This gives a range of exponents from

$$-1022 = 1 - 1023 \quad \text{to} \quad 1023 = 2046 - 1023$$

- Quad-precision floating point numbers are 128 bits wide. Their exponent are 15 bits wide and written as biased numbers in the range 1 to $32,766 = 2^{15} - 1$. The bias is $b = 16,383 = 2^{14} - 1$. This gives a range of exponents from

$$-16,382 = 1 - 16,383 \quad \text{to} \quad 16,383 = 32,766 - 16,383$$

In this way, biased numbers in the range 1 to $2^m - 2$ represent integers n between $2 - 2^{m-1}$ and $2^{m-1} - 1$.

The exponent values 0 and 255, 2047, 32767 have special meanings in these three IEEE 754 arithmetics.

$$2 - b \leq n \leq b - 1$$

Normalized fractional part

The fractional part is called the *mantissa*.

The four fraction bits $f = f_{-1}f_{-2}f_{-3}f_{-4}$ encode the natural numbers from 0 to 15, but their value is interpreted as

$$\frac{f}{16} = \frac{(f_{-1}f_{-2}f_{-3}f_{-4})_2}{2^4}$$

Finally, except for 0, which is $(0\ 000\ 0000)_{fp}$, assume that all floating point numbers are *normalized*.

This is similar to writing the decimal number 0.3141 as $3141/10^4$.

Therefore, if

$$t = s\ e_2e_1e_0\ f_{-1}f_{-2}f_{-3}f_{-4} = s\ e\ f$$

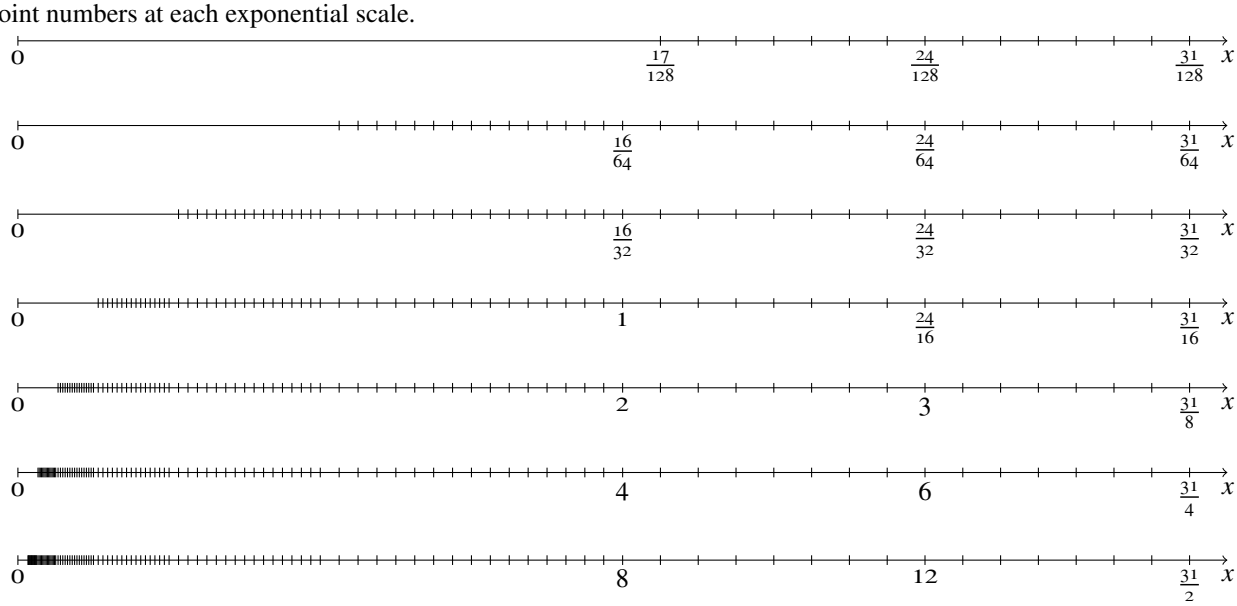
is a floating point number, it represents the rational number

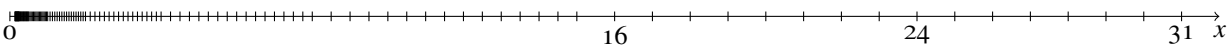
$$t = (-1)^s\left(1 + \frac{f}{16}\right) \times 2^e$$

where $s \in \mathbb{B}$, $0 \le f \le 15$, and $-3 \le e - 3 \le 4$. Here are some examples of the my pidgin encoding of 8-bit wide normalized floating point numbers.

Floating point	Decimal value	
0 000 0000	0	
1 000 0000	$-1 \cdot 1.0 \times 2^{-3}$	$-1/8$
0 001 0000	$+1 \cdot 1.0 \times 2^{-2}$	$1/4$
0 001 0001	$+1 \cdot \left(1 + \frac{1}{16}\right) \times 2^{-2}$	$17/64$
0 100 0000	$+1 \cdot 1.0 \times 2^{1}$	2
0 100 0001	$+1 \cdot \left(1 + \frac{1}{16}\right) \times 2^{1}$	$17/8$
0 111 0000	$+1 \cdot 1.0 \times 2^{4}$	16
0 111 1111	$+1 \cdot \left(1 + \frac{15}{16}\right) 0 \times 2^{4}$	31

The line graphs that follow show the distribution and density of these floating point numbers at each exponential scale.





In practice, IEEE 754 fractions can be *single*, *double*, or *quad*-precision.

- Single-precision floating point numbers are 32 bits wide. Their fractions are 23 bits wide and their *precision* is 24. Since

$$2^{23} = (2^{10})^{2.3} \approx (10^3)^{2.3} = 10^{6.9}$$

single-precision arithmetic give about 6 or 7 digits of accuracy.

- Double-precision floating point numbers are 64 bits wide. Their fractions are 52 bits wide and their *precision* is 53. Since

$$2^{52} = (2^{10})^{5.2} \approx (10^3)^{5.2} = 10^{15.6}$$

double-precision arithmetic give about 15 or 16 digits of accuracy.

- Quad-precision floating point numbers are 128 bits wide. Their fractions are 112 bits wide and their *precision* is 113. Since

$$2^{112} = (2^{10})^{11.2} \approx (10^3)^{11.2} = 10^{33.6}$$

quad-precision arithmetic give about 33 or 34 digits of accuracy.

Floating-point arithmetic

Floating-point number systems are rife with arithmetic errors. Even the most basic rules do not apply.

- Floating-point arithmetic is *not closed* under addition. For instance,

$$\text{Let} \quad t = 16 = 1.0 \times 2^4 = (0\ 111\ 0000)_{fp}$$

and

$$s = 0.5 = 1.0 \times 2^{-1} = (0\ 010\ 0000)_{fp}$$

Both t and s are pidgin floating point numbers. But, their sum $t + s = 16.5$ cannot be written as a pidgin floating point number.

$$\begin{aligned} 16 + 0.5 &= (1.0)_2 \times 2^4 + (1.0)_2 \times 2^{-1} \\ &= (1.0)_2 \times 2^4 + (0.00001)_2 \times 2^4 \\ &= (1.00001)_2 \times 2^4 \end{aligned}$$

Five bits after the binary point are needed.

$$t + s = 16.5 = 33 \times 2^{-1} = (0\ 111\ 00001)_{fp}$$

- Floating-point arithmetic is *not associative*. For instance

$$(16 + 0.5) + 0.5 = 16 \quad \text{but} \quad 16 + (0.5 + 0.5) = 17$$

Addition is computed by

- Align exponents to the higher power: In this case, float the exponent of s to 4. That is,

$$s = 1.0 \times 2^{-1} = 0.00001 \times 2^4$$

- Add fractional parts

$$t+s = 1.00000\times2^4+0.00001\times2^4 = 1.00001\times2^4$$

- Truncate to the precision of the system: 4 places to the right of the binary point: $16 + 0.5 = 1.0000 \times 2^4 = 16$.

The floating-point number closest to $x = 8/128 = 1/16$ is $t = 0$. Therefore $1/16$ rounds to 0. The rounding error is $e(1/16) = 1/16$ and the relative error is $E(1/16) = 1$.

The floating point number closest to $x = 16/128 = 1/8$ is $t = 17/128$. Therefore $1/8$ rounds to $17/128$. The rounding error is $e(1/8) = 1/128$ and the relative error is $E(1/8) = 1/16$.

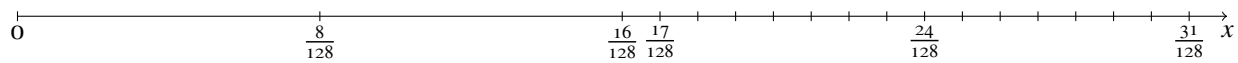
The floating point number closest to $x = 33/256$ is $t = 17/128$. Therefore $33/256$ rounds to $17/128$. The rounding error is $e(33/256) = 1/256$ and the relative error is $E(33/256) = 1/33$.

There are several measures of floating point errors that occur when approximating arithmetic over the real numbers.

Let $t(x)$ be the floating point number closest to real number x. Then $e(x) = |x - t(x)|$ is the *rounding error* in approximating x by floating point number $t(x)$. As the graphs above show, the rounding error increases as the magnitude of x increases. The *relative error* in approximating $x \neq 0$ by $t(x)$ is defined to be

$$E(x) = \frac{e(x)}{|x|} = \frac{|x - t(x)|}{|x|}$$

Look at the first line graph above, where the magnitude of scale is the smallest.



Example: Real numbers approximated by floating point numbers

Real x	*Floating point* $t(x)$	*Rounding error* $e(x)$	*Relative error* $E(x)$
$\frac{8}{128}$	0	$\frac{1}{16}$	1
$\frac{16}{128}$	$\frac{17}{128}$	$\frac{1}{128}$	$\frac{1}{16}$
$\frac{33}{256}$	$\frac{17}{128}$	$\frac{1}{256}$	$\frac{1}{33}$
$\frac{33}{128}$	$\frac{16}{64}$	$\frac{1}{128}$	$\frac{1}{33}$
$\frac{33}{64}$	$\frac{16}{32}$	$\frac{1}{64}$	$\frac{1}{33}$
$\frac{33}{32}$	$\frac{16}{16}$	$\frac{1}{32}$	$\frac{1}{33}$
$\frac{33}{16}$	$\frac{16}{8}$	$\frac{1}{16}$	$\frac{1}{33}$
$\frac{33}{8}$	$\frac{16}{4}$	$\frac{1}{8}$	$\frac{1}{33}$
$\frac{33}{4}$	$\frac{16}{2}$	$\frac{1}{4}$	$\frac{1}{33}$
$\frac{33}{2}$	$\frac{16}{1}$	$\frac{1}{2}$	$\frac{1}{33}$

Machine epsilon

The machine epsilon ϵ for floating-point arithmetic is defined to be the largest power of the base such that

$$1 + \epsilon = 1$$

$1 = 1 + 2^{-6} = 65/64 \notin \mathbb{Q}_{fp}$
$1 = 1 + 2^{-5} = 33/32 \notin \mathbb{Q}_{fp}$
$1 < 1 + 2^{-4} = 17/16 \in \mathbb{Q}_{fp}$

In a binary floating point system, the machine epsilon will be a power of 2. In our pidgin floating-point arithmetic, $\epsilon = 2^{-5}$. Notice that

$$1 = 1 + 2^{-5} = 33/32 = (1.00001)_2 \quad \text{rounds to} \quad (1.0000)_2$$

The *machine's epsilon* measures the *precision* (accuracy or inaccuracy) of floating-point arithmetic. Machine epsilon can be computed by the formula

$$\epsilon = 2^{-p} \quad \text{where } p \text{ is the precision of the system.}$$

In practice, the machine epsilon for IEEE 754 *single*, *double*, or *quad*-precision numbers are:

- Single-precision floating point numbers are 32 bits wide with a precision of 24. Single-precision machine epsilon is

$$2^{-24} \approx 5.96046447754 \times 10^{-8}$$

- Double-precision floating point numbers are 64 bits wide. with a precision of 53. Double-precision machine epsilon is

$$2^{-53} \approx 1.11022302463 \times 10^{-16}$$

- Quad-precision floating point numbers are 128 bits wide. with a precision of 113. Quad-precision machine epsilon is

$$2^{-113} \approx 9.62964972194 \times 10^{-35}$$

Pass a Quiz: Floating point notation

Take a quiz on page 322 to check your understanding. You can return to here from the quiz.

Homework Questions

🏠 Use your time outside of class to solve these problems.

1 Use Horner's to convert the following unsigned binary numbers to decimal.

1.1 $(1001001)_2$

1.2 $(0001001)_2$

1.3 $(10010001)_2$

1.4 $(011110)_2$

2 Use *repeated remaindering* to convert the following unsigned integers to binary.

2.1 73

2.2 37

2.3 237

2.4 105

3 I got this one from Click & Clack, the Tappit Brothers: Given \$1000 in \$1 dollar bills and 10 envelops, distribute the money among the envelops so that you can give out any dollar amount from \$1 to \$1000.

4 Convert the following two's complement (signed) integers to decimal notation.

4.1 $(1001001)_{2c}$

4.2 $(0001001)_{2c}$

4.3 $(10010001)_{2c}$

4.4 $(011110)_{2c}$

5 Convert the following signed integers to two's complement notation.

5.1 -73

5.2 $+37$

5.3 -237

5.4 $+105$

6 Convert the following fixed-point binary numbers to decimal fractions.

6.1 $(0.011101)_2$

6.2 $(100.111101)_2$

7 Convert the following decimal numbers to fixed-point binary fractions.

7.1 0.575

7.2 57.5

8 Write your answers from problem (7) in *normalized* floating-point notation.

9 Using the 8-bit pidgin floating-point notation described in these notes, covert the following to their decimal representation.

9.1 $(1\ 000\ 1101)_{fp}$

9.2 $(0\ 000\ 1101)_{fp}$

9.3 $(1\ 111\ 0000)_{fp}$

9.4 $(1\ 111\ 0001)_{fp}$

10 Consider normalized single precision IEEE 754 floating point numbers.

10.1 What string of bits represents the positive number closest to zero?

10.2 Write this string using hexadecimal numerals.

10.3 What is the value of this number?

10.4 What string of bits represents the positive number farthest from zero?

10.5 Write this string using hexadecimal numerals.

10.6 What is the value of this number?

9. Names: By any name

God taught Adam the names of all things.

Bible and Torah, Genesis, chapter 2, verse 19. Quran, Sura Al-Baqarah, verse 31

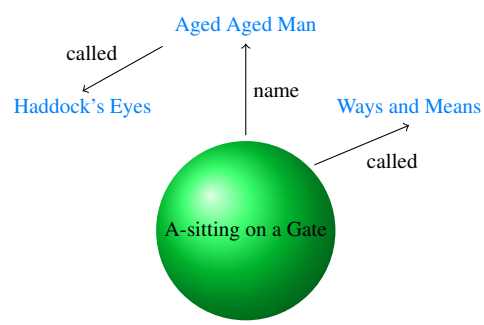
Mathematics is the art of giving the same name to different things.

Henri Poincaré, The Future of Mathematics

What's in a name? That which we call a rose by any other name would smell as sweet.

William Shakespeare, Romeo and Juliet, Act II, Scene I

"The name of the song is called 'Haddock's Eyes'." "Oh, that's the name of the song, is it?" Alice said, trying to feel interested. "No, you don't understand," the Knight said, looking a little vexed. "That's what the name is called. The name really is 'The Aged Aged Man'." "Then I ought to have said 'That's what the song is called?'" Alice corrected herself. "No, you oughtn't: that's quite another thing! The song is called 'Ways and Means': but that's only what it's called, you know!" "Well, what is the song, then?" said Alice, who was by this time completely bewildered. "I was coming to that," the Knight said. "The song really is 'A-sitting on a Gate': and the tune's my own invention." (Carroll et al., 2000)



Every thing should have a name. Through translation, every *thing* chosen from a *countable* discrete set can be named by one of the natural numbers.

$$0,\ 1,\ 2,\ 3, \ldots$$

So a *thing* can be named by a number n, where n itself is the name of a *number thing*. The *thing* itself is *called* by some name: "God," "Adam," "Juliet," "Montague," "Mathematics," "Art," Lastly, the *thing* has a *value* which really is the *thing*.

The set of natural numbers is named by the symbol $\mathbb{N}$.

$$\mathbb{N} = \{0,\ 1,\ 2,\ 3, \ldots\}$$

Some people do not include 0 as a natural numbers. But, nothing is important! Nothing is more natural than nothing! The opposite of nothing is everything. Everything is important!

A single thing can have more than one name. For instance, $(41)_{10}$ and $(0010\ 1001)_2$ are both names for the same natural number called "forty-one." Multiple names for the same *value* are called *aliases*. When a thing has *aliases*, algorithms that convert between them are useful.

Alphabets

There are several commonly used alphabets.

- The alphabet of bits $\mathbb{B} = \{0, 1\}$ for writing numbers in binary.

- The alphabet of digits

$$\mathbb{D} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} = \mathbb{B} \cup \{2, 3, 4, 5, 6, 7, 8, 9\}$$

  for writing numbers in decimal.

- The alphabet of hexadecimal numerals

$$\mathbb{H} = \{0, 1, 2, 3, \ldots, 7, 8, 9, A, B, C, D, E, F\} = \mathbb{D} \cup \{A, B, C, D, E, F\}$$

  for writing in hexadecimal.

Hexadecimals are frequently used in describing computer organization: How a machine is built and operated. Binary numbers are too long. Hexadecimal numbers have just about the right length for recognizing patterns.

There are also alphabets used for natural language and symbols. Instance are:

- The 26 letters in the lowercase English alphabet.

$$\mathbb{E} = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z\}$$

  These letters are encoded in Unicode as decimal 97 through 122 or their hexadecimal equivalent:

$$\mathbb{E} = \{(60)_{16}, (61)_{16}, (62)_{16}, \ldots, (78)_{16}, (79)_{16}, (7A)_{16}\}$$

- The 24 letters in the Greek alphabet is commonly used in mathematics.

The Greek alphabet is also used by those who write in Greek.

$$\mathbb{G} = \{\alpha, \beta, \gamma, \delta, \epsilon, \zeta, \eta, \theta, \iota, \kappa, \lambda, \mu, \nu, \xi, o, \pi, \rho, \sigma, \tau, \upsilon, \phi, \chi, \psi, \omega\}$$

  These 24 lower case Greek letters are encoded in Unicode as decimal 945 through 968 or their hexadecimal equivalent:

$$\mathbb{G} = \{(3B1)_{16}, (3B2)_{62}\ 16, (3B4)_{16}, \ldots, (3C6)_{16}, (3C8)_{16}, (3C8)_{16}\}$$

- The Unicode character set is meant to be universal: Inclusive of all symbols used in all languages. The Unicode set of characters have decimal names from 0 to 1, 114, 111 or equivalent hexadecimal names

$$\mathbb{U} = \{c : 0 \leq c \leq (10FFFF)_{16} = 1, 114, 111\}$$

Strings

A string over an alphabet $\mathbb{A}$ is a sequence of characters from $\mathbb{A}$. There is one empty string λ that names nothing; it contains no characters; its length is 0, $|\lambda| = 0$. For illustration, let $\mathbb{A} = \mathbb{E}$ the lower case English alphabet. Then, the set of all strings over $\mathbb{E}$ is

$\mathbb{E}^*$ is call the Kleene closure of $\mathbb{E}$.

$$\mathbb{E}^* = \{\lambda, a, b, c, \ldots, z, aa, \ldots az, \ldots, za, \ldots, zz, aaa, \ldots\}$$

Naming numbers

A fundamental naming problem is:

How many numerals are needed to write the number m?

To solve the problem in the binary case, notice that any number $m \geq 1$ can be bound below and above by powers of 2

$\mathbb{N}^+ = \{1, 2, 3, \ldots\}$ is the set of positive integers.

$$(\forall\, m \in \mathbb{N}^+)(\exists n \in \mathbb{N})(2^n \leq m < 2^{n+1})$$

For instance,

$$\begin{aligned}
&1 = 2^0 \leq 1 < 2^1 = 2\\
&2 = 2^1 \leq 2,\ 3 < 2^2 = 4\\
&4 = 2^2 \leq 4,\ 5,\ 6,\ 7 < 2^3 = 8\\
&8 = 2^3 \leq 8,\ 9,\ A,\ B,\ C,\ D,\ E,\ F < 2^4 = 16
\end{aligned}$$

and so on. Furthermore, to write 2^n in binary requires $(n+1)$ bits.

$$\begin{aligned}
2^0 &= (1)_2\\
2^1 &= (10)_2\\
2^2 &= (100)_2\\
2^3 &= (1000)_2
\end{aligned}$$

and so on. Therefore, in the general case,

to write m in the range $2^n \leq m < 2^{n+1}$ requires $n+1$ bits.

You can compute $(n+1)$ using the inequalities $2^n \leq m < 2^{n+1}$. Take the log base 2 of each term to deduce that

$$n \leq \lg m < n + 1$$

Take the floor of $\lg m$ to get n and add 1 to get $n+1$, the number of bits needed to write m.

$$n + 1 = \lfloor \lg m \rfloor + 1 = \text{number of bits to write } m.$$

In the general case, any number $m \geq 1$ can be bound below and above by powers of b, for any numerical base $b > 1$.

$$(\forall\, m \in \mathbb{N}^+)(\exists n \in \mathbb{N})(b^n \leq m < b^{n+1})$$

For instance,

$$\begin{aligned}
&1 \leq 1,\ 2, \ldots,\ b-1 < b\\
&b \leq b,\ b+1, \ldots,\ b^2-1 < b^2\\
&b^2 \leq b^2,\ b^2+1,\ \ldots,\ b^3-1 < b^3
\end{aligned}$$

and so on.

When $b^n \leq m < b^{n+1}$, $n+1$ numerals are needed to write m in base b. Take the log base b of each term, to deduce that

$$n \leq \log_b m < n+1$$

Take the floor of $\log_b m$ to get n and add 1 to get $n+1$, the number of base b numerals needed to write m.

$$\lfloor \log_b m \rfloor + 1 = \text{number of base } b \text{ numerals needed to write } m.$$

Theorem 11: String length of a number

Let $\mathbb{A}$ be a number system with cardinality $b = |\mathbb{A}|$. To write the number m in base b notation requires

$$n + 1 = \lfloor \log_b m \rfloor + 1 = \text{ numerals.}$$

Numerals needed to represent a number

Number represented m	Bits needed $\lfloor \lg m \rfloor + 1$	Number represented m	Digits needed $\lfloor \log m \rfloor + 1$
$1 \leq m < 2$	1	$1 \leq m < 10$	1
$2 \leq m < 4$	2	$10 \leq m < 100$	2
$4 \leq m < 8$	3	$100 \leq m < 1000$	3
$8 \leq m < 16$	4	$1000 \leq m < 10^4$	4
$16 \leq m < 32$	5	$10^4 \leq m < 10^5$	5

Pass a Quiz: Naming Numbers

Take a quiz on page 323 to check your understanding. You can return to here from the quiz.

Counting strings

The number of strings that can be written depends on the cardinality of the alphabet and the *string length*. The length can be *fixed*, say n for some fixed, but unspecified natural number. Or, the length can *vary* from, say $k = 0$ to $k = (n-1)$.

Fixed length strings

There are several examples that you have mastered.

- You can write 2^n different binary strings of (fixed) length n: $2^1 = 2$ strings of length 1, $2^2 = 4$ strings of length 2, $2^3 = 8$ strings of length 3, etc.

- You can write 10^n different decimal strings of (fixed) length n: $10^1 = 10$ strings of length 1, $10^2 = 100$ strings of length 2, $10^3 = 1000$ strings of length 3, etc.

- You can write 16^n different hexadecimal strings of (fixed) length n: $16^1 = 16$ strings of length 1, $16^2 = 256$ strings of length 2, $16^3 = 4096$ strings of length 3, etc.

- In general, there are $|\mathbb{A}|^n$ different strings of length n over an alphabet $\mathbb{A}$ with cardinality $|\mathbb{A}|$.

Pretend you want to name m things using using fixed-length strings over an alphabet $\mathbb{A}$ with $|\mathbb{A}| = c$. You need to compute the string length n. You can construct a function that maps m and b to n.

With little loss of generality, you can assume the names are $0, 1, 2, \ldots, (m-1)$. This *reduces* the problem of computing the number of base b numerals needed to write m names to the problem of computing the number of base b numerals needed to write the number$(m-1)$.

If you've studied the previous notes, you will realize that $\log_b(m-1)$ is a good approximation the string length needed to name m things. You know any number up to $b^n - 1$ can be written in n base b numerals. Therefore, you need to compute the least integer exponent n such that

$$b^{n-1} \leq (m-1) < b^n \quad \text{or, if you prefer} \quad b^{n-1} < m \leq b^n$$

You can compute n by taking the logarithm base b of all sides in the above inequality, and then computing the floor of the log.

$$\begin{aligned} b^{n-1} &\leq (m-1) < b^n \\ n-1 &\leq \log_b(m-1) < n \\ n-1 &= \lfloor \log_b(m-1) \rfloor < n \\ n-1 &< \lfloor \log_b(m-1) \rfloor + 1 = n \end{aligned}$$

A more simple formula for n is $n = \lceil \log_b m \rceil$. Here's how you can show

$$\lfloor \log_b(m-1) \rfloor + 1 = \lceil \log_b m \rceil$$

$$\begin{aligned} b^{n-1} \leq (m-1) < b^n \quad &\Rightarrow \quad b^{n-1} < m \leq b^n \\ b^{n-1} < m \leq b^n \quad &\Rightarrow \quad (n-1) < \log_b m \leq n \\ (n-1) < \log_b m \leq n \quad &\Rightarrow \quad (n-1) < \lceil log_b m \rceil = n \end{aligned}$$

For instance, you can name 37 different things using binary strings of length

$$\lceil \lg 37 \rceil = \lceil 5.20945336562\cdots \rceil = 6$$

You know this because binary strings of length 5 can only name $2^5 = (10\,0000)_2 = 32$ things, while 6 bits strings can name $2^6 = (100\,0000)_2 = 64$ things. In fact, you can write

$$37 = (10\,0101)_2$$

As another example, using the quaternary alphabet $\{0, 1, 2, 3\}$, you can make 37 different names with strings of length

$$\lceil \log_4 37 \rceil = \lceil 2.60472668281\cdots \rceil = 3$$

You know this because quaternary strings of length 2 can only name $4^2 = (100)_4 = 16$ things, while 4 "quatits" strings can name $4^3 = (1000)_4 = 64$ things. In fact, you can write

$$37 = (211)_4$$

For instance, in decimal, any number $(m-1) = 1000, 1001, \ldots, 9999$, that is any m such that

$$10^3 \leq (m-1) < 10^4$$

can can written in 3 digits. *Reduction* of a new problem on one you already know how to solve is an important problem-solving technique.

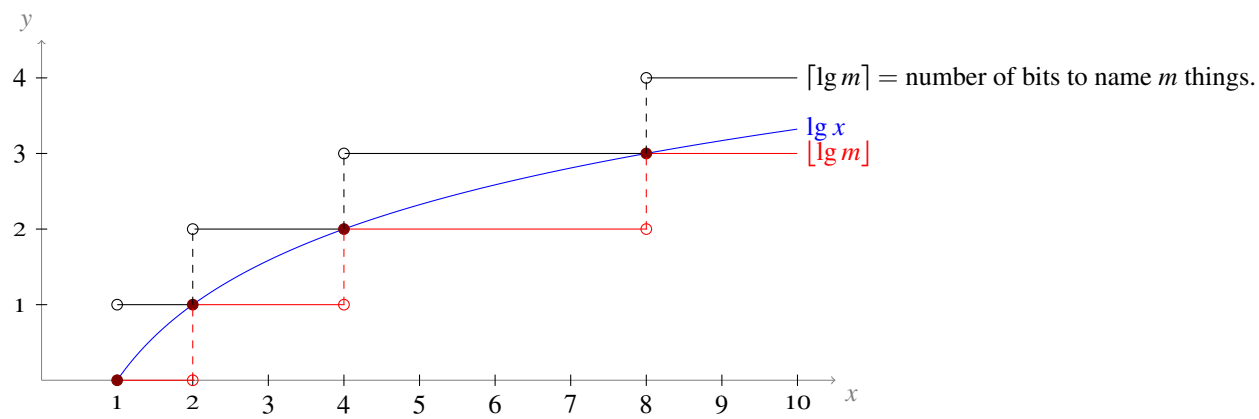
Theorem 12: Counting fixed-length strings

Let $\mathbb{A}$ be an alphabet with cardinality $b = |\mathbb{A}|$. To name m things using fixed-length strings over $\mathbb{A}$ requires their length to be $n = \lceil \log_c m \rceil$.

$$n = \lceil \log_b m \rceil = \text{ string length needed to name } m \text{ things.}$$

Look at the the graph below. The top curve $y = \lceil \lg x \rceil$ describes the number of bits to name m things. Notice that to name:

- 2 things requires 1 bits;
- 3 or 4 things requires 2 bits;
- 5, 6, 7, or 8 things requires 3 bits.



Variable length names

Letting the length of strings change give rise to a useful idea: Geometric sums, some call them *geometric progressions*. Consider the binary, decimal, and hexadecimal alphabets.

- Use the binary alphabet $\mathbb{B}$ and vary the string length from 0 to $(n-1)$. You can see that there are

  $$1+2+4+8+\cdots+2^{n-1} = 2^n - 1 \quad \text{different variable length binary strings.}$$

  You know this because $1+2+4+8+\cdots+2^{n-1}$ is the binary number of all 1's

  $$\left(\overbrace{11\cdots11}^{n\text{-bits}}\right)_2 = 2^n - 1$$

  Recall, $(11)_2 = 3 = 2^2 - 1$, $(111)_2 = 7 = 2^3 - 1$, $(1111)_2 = 15 = 2^4 - 1$, etc.

- Use the decimal alphabet $\mathbb{D}$ and vary the string length from 0 to $(n-1)$.
  You can compute that there are

  $$1+10+100+1000+\cdots+10^{n-1} = \frac{10^n-1}{9} \quad \text{different variable length decimal strings.}$$

  You know this because $1+10+100+1000+\cdots+10^{n-1}$ is the decimal string of all 1's

  $$\left(\overbrace{11\cdots11}^{n\text{-digits}}\right)_{10} = \frac{(99\cdots99)_{10}}{9}$$

  Recall, $99 = 10^2-1$, $999 = 10^3-1$, $9999 = 10^4-1$, etc.

- Use the hexadecimal alphabet $\mathbb{H}$ and vary the string length from 0 to $(n-1)$.
  You can compute that there are

  $$1+16+256+4096+\cdots+16^{n-1} = \frac{16^n-1}{15} \quad \text{different variable length hexadecimal strings.}$$

  You know this because $1+16+256+4096+\cdots+16^{n-1}$ is the hexadecimal string of all 1's

  $$\left(\overbrace{11\cdots11}^{n\text{-hexits}}\right)_{16} = \frac{(FF\cdots FF)_{16}}{F}$$

  Recall, $FF = 16^2-1$, $FFF = 16^3-1$, $FFFF = 16^4-1$, etc.

Now consider the general question.

How many names that can be written using alphabet $\mathbb{A}$ with cardinality b if the string length varies from 0 to $(n-1)$?

Let's use the $\mathbb{DNA}$ alphabet as an example.

$$\mathbb{DNA} = \{A, C, G, T\}$$

The strings over $\mathbb{DNA}$ are:

$\langle \lambda, A, C, G, T, AA, AC, AG, AT, CA, CC, CG, \ldots \rangle$

There is 1 string of length 0, 4 of length 1, $4^2 = 16$ strings of length 2, $4^3 = 64$ strings of length 3, and so on. To count all of these strings up to some limit, say $(n-1)$, compute the value of the sum

$$1+4+4^2+\cdots+4^{n-1} = \sum_{k=0}^{n-1} 4^k = \frac{4^n-1}{3}$$

Don't like naming things by nothing? Subtract 1 from the sum to get

$$4+4^2+\cdots+4^{n-1} = \sum_{k=1}^{n-1} 4^k = \frac{4^n-1}{3}-1 = \frac{4^n-4}{3}$$

as the number of non-empty $\mathbb{DNA}$ variable-length strings of length $(n-1)$ or less.

If you write the numbers $\frac{4^n-1}{3}$ in *quaternary* (base 4) notation for $n = 0, 1, 2, 3, 4, \ldots$ you will generate the sequence

$$(0)_4, (1)_4, (11)_4, (111)_4, (1111)_4, \ldots$$

If you write the numbers $\frac{4^n-4}{3}$ in *quaternary* notation for $n = 1, 2, 3, 4, 5, \ldots$ you will generate the sequence

$$(0)_4, (10)_4, (110)_4, (1110)_4, (11110)_4, \ldots$$

You know

- There is 1 string of length 0,
- There are b strings of length 1,
- There are b^2 strings of length 2,
- And so on, up to b^{n-1} strings of length $(n-1)$.

The geometric sum

$$1+b+b^2+\cdots+b^{n-1} = \sum_{k=0}^{n-1} b^k = \frac{b^n-1}{b-1}, \quad b \neq 1$$

computes the answer to the question.

There are several ways to interpret the equivalence

$$\sum_{k=0}^{n-1} b^k = \frac{b^n-1}{b-1}$$

One way to see this is to recognize that $b-1$ is the largest natural number that can be written using a single base b numeral. And, using positional notation, you know

$$(b-1)(1+b+b^2+\cdots+b^{n-1}) = (b-1)\sum_{k=0}^{n-1} b^k = b^n - 1$$

is the largest base b number that can be written with n numerals.

You can also use concepts from the chapter on the sum & difference calculus. Note that b^k can be written as the *difference*.

$$b^k = \frac{b^k(b-1)}{(b-1)} = \frac{b^{k+1}}{b-1} - \frac{b^k}{b-1}$$

Therefore, the sum *telescopes*.

$$\begin{aligned}
1+b+\cdots+b^{n-1} &= \left(\frac{b}{b-1} - \frac{1}{b-1}\right) + \left(\frac{b^2}{b-1} - \frac{b}{b-1}\right) + \cdots + \left(\frac{b^n}{b-1} - \frac{b^{n-1}}{b-1}\right) \\
&= \frac{b^n}{b-1} - \frac{1}{b-1} \\
&= \frac{b^n-1}{b-1}
\end{aligned}$$

You can also see the formula for a geometric sum is correct by algebraic manipulation.

$$\begin{aligned}
(b-1)(1+b+b^2+\cdots+b^{n-1}) &= (b+b^2+b^3+\cdots+b^n) - (1+b+b^2+\cdots+b^{n-1}) \\
&= b^n - 1
\end{aligned}$$

Here are some examples.

- In binary you can write $2^3 - 1 = 7 = (111)_2$ variable-length strings of length 2 or less

  $$\langle \lambda, 0, 1, 00, 01, 10, 11 \rangle$$

  There are $2^3 - 2 = 6$ non-empty binary strings of length 1 or 2.

- In decimal, you can write $(10^3 - 1)/9 = 111$ variable length strings of length 2 or less

  $$\langle \lambda, 0, \cdots, 9, 00, 01, \ldots, 09, 10, \ldots, 19, \ldots, 98, 99 \rangle$$

  One of length 0, 10 of length 1, and 100 of length 2. There are $(10^3 - 10)/9 = 990/9 = 110$ non-empty decimal strings of length 1 or 2.

- In hexadecimal, you can write $(16^3 - 1)/15 = (111)_{16} = 273$ variable length strings of length 2 or less

  $$\langle \lambda, 0, \cdots, F, 00, 01, \ldots, 0F, 10, \ldots, 1F, \ldots, FE, FF \rangle$$

  One of length 0, 16 of length 1, and 256 of length 2. There are $(16^3 - 16)/15 = (FF0/F)_{16} = (110)_= 256 + 16 = 272$ non-empty hexadecimal strings of length 1 or 2.

- And, finally, for the DNA alphabet, there are $(4^3 - 1)/3 = 21$ variable-length strings of length 2 or less

  $\langle \lambda, A, C, G, T, AA, AC, AG, AT, CA, CC, CG, CT, GA, GC, GG, GT, TA, TC, TG, TT \rangle$

  One of length 0, 4 of length 1, and $4^2 = 16$ of length 2. There are $(4^3 - 4)/3 = 60/3 = 20$ non-empty quaternary strings of length 1 or 2.

In general, for variable-length strings, you know there are $(b^n - b)/(b - 1)$ non-empty strings whose length varies from 1 to $(n - 1)$. So to name m things, you need to compute the least integer n such that

$$m \leq \frac{b^n - b}{b - 1}$$

To solve for n requires a little more dexterity.

$$\begin{aligned} m &\leq \frac{b^n - b}{b - 1} \\ m(b-1) &\leq b^n - b \\ m(b-1) + b &\leq b^n \\ \log_b(m(b-1) + b) &\leq n \\ \lceil \log_b(m(b-1) + b) \rceil &= n \end{aligned}$$

Consider using variable-length non-empty binary strings. To name m things requires strings of length up to $\lceil \lg(m + 2) \rceil - 1$ bits.

Naming things in binary

Fixed-length strings		Variable-length strings	
Things named m	Bits needed $\lceil \lg m \rceil$	Things named m	Bits needed $\lceil \lg (m + 2) \rceil - 1$
$1 < m \leq 2$	1	$2 < m \leq 6$	2
$2 < m \leq 4$	2	$6 < m \leq 14$	3
$4 < m \leq 8$	3	$14 < m \leq 30$	4
$8 < m \leq 16$	4	$30 < m \leq 62$	5
$16 < m \leq 32$	5	$62 < m \leq 126$	6

Pass a Quiz: Naming Basics

Take a quiz on page 323 to check your understanding. You can return to here from the quiz.

Homework Questions

🏠 Use your time outside of class to solve these problems.

1 How many bits are needed to write the following numbers in binary? For small numbers, you may be able to see the answer in your head. For larger numbers use appropriate functions to compute the result.

1.1 32

1.2 73

1.3 255

1.4 256

1.5 737

1.6 37373737

1.7 If these numbers are written in hexadecimal notation, how many numerals are needed?

2 What is the general formula for the number of numerals needed to write a natural number n in base b notation?

3 What is the general formula for the number of bits needed to write an integers n two's complement notation?

4 Pretend you wanted to name things using natural numbers: $0, 1, \ldots, (m-1)$. How many bits are needed to name the following number of things.

4.1 32

4.2 73

4.3 255

4.4 256

4.5 737

4.6 37373737

4.7 If names are written in hexadecimal notation, how many numerals are needed to names the given number of things?

5 What is the general formula for the number of numerals needed to name n using numbers written in base b notation?

6 The Arabic alphabet includes 28 letters. How many Arabic strings of length $(n-1)$ are there? How many Arabic strings are there of length k for $k = 0, 1, \ldots, (n-1)$, that is, sum the individual results?

7 Think of seven plus or minus two things. How many bits are needed to name 5, 6, 7, 8 or 9 things? (Miller, 1994)

8 Suppose you had to name many things. Which would be easier for you? Explain why you made your choice.

Use names such as "ArrowBoldDownRight," "ArrowBoldRightCircled," etc.

Use an array, say DING[], where each entry points to one of these names, for instance DING[123]="ArrowBoldDownRight?"

9 For a modern perspective on alphabets: How many symbols (characters) are in the Unicode alphabet? How are they organized?

10 For historical perspective: How many symbols are in the ASCII alphabet? How many symbols are in the EBCDIC alphabet?

11 I've drawn the graph of the number of bits needed to write m as a constant step function, as m goes from 2^{n-1} to $2^n - 1$, but the graph only fits the data the natural numbers. For instance, to write $2.5 = 5/2$ requires 3 bits

$$(10.1)_2 = 2.5$$

More than $\lfloor \lg 2.5 \rfloor + 1 = 2$, which the graph suggests. Show the following.

11.1 Let $x = n \times 2^m$, how many bits are needed to write x?

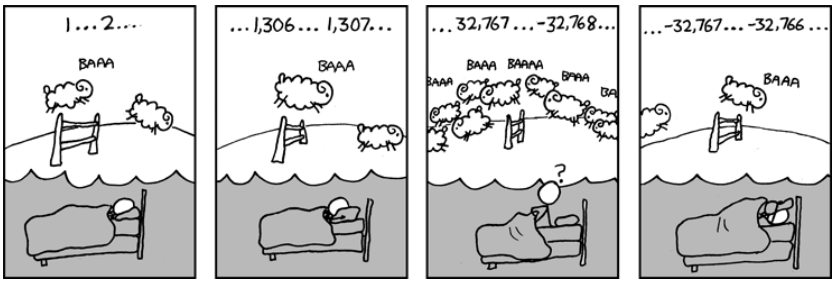
11.2 If $x \neq n/2^m$, show that the the binary representation of x is infinitely long.

11.3 Suggest how to draw the *correct* graph of the number of bits needed to write x.

Pass a Quiz: Summative exam #3 on machine numbers and naming things

Take a quiz on page 325 to check your understanding. You can return to here from the quiz.

10. Counting: How many are there?



Can't Sleep

Everything that can be counted does not necessarily count; everything that counts cannot necessarily be counted.

Albert Einstein

Counting is something we learn so early in life that we tend to dismiss it as a trivial skill, beneath the notice of mathematics.

Brian Hayes, (Hayes, 2001)

Counting is fundamental. There are too many things that count to count them all. Here are a few that I think are useful.

The 50¢ word for the study of counting is combinatorics.

Counting truth assignments

A single Boolean variable P has two (2^1) truth assignments.

0 is shorthand for `False`. Some use F and some use $\perp$ for `False`.
1 is shorthand for `True`. Some use T and some use $\top$ for `True`.

1. One `truth assignment`: $\mathrm{P} = 0$
2. Another `truth assignment`: $\mathrm{P} = 1$

The set $\mathbb{B} = \{0, 1\}$ of bits name all truth assignments on 1 Boolean variable.

A pair of Boolean variables P and Q have four (2^2) truth assignments.

1. One `truth assignment`: $\mathrm{P} = 0$, $\mathrm{Q} = 0$
2. Another `truth assignment`: $\mathrm{P} = 0$, $\mathrm{Q} = 1$

3. A third `truth assignment`: $\mathrm{P} = 1$, $\mathrm{Q} = 0$

4. And a fourth `truth assignment`: $\mathrm{P} = 1$, $\mathrm{Q} = 1$

The set

$$\mathbb{B}^2 = \mathbb{B} \times \mathbb{B} = \{(0, 0), (0, 1), (1, 0), (1, 1)\} \tag{9}$$

$\mathbb{B} \times \mathbb{B}$ is a Cartesian product, named for René Descartes, who existed because he thought: "*Je pense, donc je suis.*"

lists all truth assignments on two Boolean variables P and Q.

Taking an inductive leap, there should be $8 = 2^3$ truth assignments on three Boolean variables: P, Q, and R: For each truth assignment on P and Q, there are 2 truth assignments for R: One where R is `False`, and one where R is `True`. The set $\mathbb{B}^3$ lists all truth assignments on 3 Boolean variables.

The cardinality of $\mathbb{B}^3$ is 2^3.

$$\mathbb{B}^3 = \mathbb{B} \times \mathbb{B} \times \mathbb{B} = \{(0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1)\} \tag{10}$$

Notice 0 and 1 have been inserted into each ordered pair from equation 9 to create the list in equation 10. This doubles the size of $\mathbb{B}^2$ to create $\mathbb{B}^3$. In general, there is this theorem:

Theorem 13: Counting truth assignments

There are 2^n truth assignments on n Boolean variables. That is,

$$|\mathbb{B}^n| = |\mathbb{B}|^n = 2^n$$

is the cardinality of n-dimensional Boolean space.

$$\begin{aligned} \mathbb{B}^n &= \overbrace{\mathbb{B} \times \cdots \times \mathbb{B}}^{n \text{ factors}} && n \geq 1 \\ &= \mathbb{B} \times \overbrace{\mathbb{B} \times \cdots \times \mathbb{B}}^{(n-1) \text{ factors}} \\ &= \bigtimes_{k=0}^{n-1} \mathbb{B} \end{aligned}$$

It is interesting to apply theorem () when $n = 0$. In this case, there is only $1 = 2^0$ truth assignment: Pessimists will say it is `False`; optimists will select `True`. There's not much room in 0-dimensional space!

Recall from the notes on arithmetic $\mathbb{Z}_n$ is the set of integers modulo n.

$$\mathbb{Z}_n = \{0, 1, 2, \ldots, (n-1)\}$$

Proof: Counting truth assignments

You've seen the theorem in `True` *for* $n = 1$, $n = 2$, *and* $n = 3$. *This establishes a* basis *for mathematical induction.*
Let's hypothesize *the theorem is* `True` *for some* n, *that is,*

$$|\mathbb{B}^n| = \left|\left\{(\mathrm{P}_0, \mathrm{P}_1, \mathrm{P}_2, \ldots, \mathrm{P}_{(n-1)}) : \mathrm{P}_k \in \mathbb{B},\ k \in \mathbb{Z}_n\right\}\right| = 2^n$$

Then, *including another variable* P_n, *which can have one of two values, you construct* $\mathbb{B}^{n+1}$ *and double the cardinality of* $\mathbb{B}^n$.

$$\left|\mathbb{B}^{n+1}\right| = |\mathbb{B} \times \mathbb{B}^n| = |\mathbb{B}| \cdot |\mathbb{B}^n| = 2 \times 2^n = 2^{n+1}$$

Counting Boolean expressions

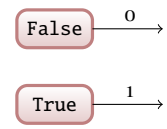
A Boolean function b maps truth assignments t to either `True` or `False`.

$$b : \mathbb{B}^n \to \mathbb{B}$$

You can count the number of Boolean functions based on n, the number of its input values.

Boolean expressions on no input variables

A constant can be thought of an function without any input. There are two ($2^{2^0} = 2^1 = 2$) Boolean expressions without input:

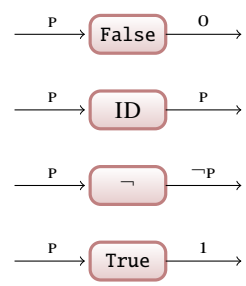


1. $b_0() =$ `False`, and

2. $b_1() =$ `True`

Boolean expressions on one variable

A function can map one Boolean variable P in four ($2^{2^1} = 2^2 = 4$) ways.



1. $b_0(\text{P}) =$ `False`

2. $b_1(\text{P}) = \text{P}$

3. $b_2(\text{P}) = \neg\text{P}$

4. $b_3(\text{P}) =$ `True`

Boolean expressions on two variables

A function $b(\text{P}, \text{Q})$ can map two Boolean variables P and Q in sixteen ($2^{2^2} = 2^4 = 16$) ways. Several of these are fundamental and useful, all of them can be named.

The numerical relations between a pair of Boolean variables P and Q are the familiar.

$=$	equality	$\leq$	less than or equal
$\neq$	not equal	$>$	greater than
$<$	less than	$\geq$	greater than or equal

The table below a programming (logic) name, an English name, and a mathematical name to these six numerical Boolean relations.

==	equivalent	$\equiv$
/=	exclusive or	$\oplus$
<	Q does not imply P	$\not\to$
<=	P implies Q	$\to$
>	P does not imply Q	$\not\to$
>=	Q implies P	$\to$

The table below shows the mapping for the six numerical relations.

Boolean (comparison) opeations

Input		Outputs					
P	Q	$=$	$\neq$	$<$	$\leq$	$>$	$\geq$
0	0	1	0	0	1	0	1
0	1	0	1	1	1	0	0
1	0	0	1	0	0	1	1
1	1	1	0	0	1	0	1
Decimal		9	6	4	13	2	11

An additional ten functions can be defined on ordered pairs of Boolean variables (P, Q).

Use the decimal value of the binary string in each column to check all $2^{2^2} = 16$ Boolean relations are listed.

Boolean opeations

Input		Outputs									
P	Q	`and`	`or`	`not` P	`True`	`False`	P	`nor`	`not` Q	Q	`nand`
0	0	0	0	1	1	0	0	1	1	0	1
0	1	0	1	1	1	0	0	0	0	1	1
1	0	0	1	0	1	0	1	0	1	0	1
1	1	1	1	0	1	0	1	0	0	1	0
Decimal		1	7	12	15	0	3	8	10	5	14

Boolean expressions on n variables

The generalization to n-input Boolean functions is this: For each truth assignment t on n Boolean variables, there are 2 mappings:

1. $t \to$ `False`

2. $t \to$ `True`

From the previous section there are 2^n truth assignments t on n Boolean variables. A Boolean function maps each of these 2^n truth assignments to either `True` or `False`.

The number of factors is the number of truth assignments. There are 2 choices for each truth assignment.

Theorem 14: Counting Boolean functions

There are

$$\overbrace{2 \times 2 \times \cdots \times 2}^{2^n \text{ factors}} = 2^{2^n}$$

Boolean functions on n Boolean variables.

Proof: Counting Boolean functions

You've seen the theorem in `True` *for* $n = 0$, $n = 1$, *and* $n = 2$: *There are* $2^{2^0} = 1$, $2^{2^1} = 4$, *and* $2^{2^2} = 16$ *Boolean functions (expressions) on these numbers. This establishes a* basis *for mathematical induction.*

Let's hypothesize *the theorem is* `True` *for some* n, *that is, there are* 2^{2^n} *Boolean functions on* n *Boolean variables for some* $n \geq 0$. *That is,*

$$|\{b : \mathbb{B}^n \to \mathbb{B}\}| = 2^{2^n} \quad \text{for some } n \geq 0.$$

Then, *including another variable* P_n, *which can have one of two values, you construct* $\mathbb{B}^{n+1}$ *and double the cardinality of* $\mathbb{B}^n$.

$$\left|\mathbb{B}^{n+1}\right| = |\mathbb{B} \times \mathbb{B}^n| = |\mathbb{B}| \cdot |\mathbb{B}^n| = 2 \times 2^n = 2^{n+1}$$

Pass a Quiz: Counting truth assignments and Boolean functions

Take a quiz on page 326 to check your understanding. You can return to here from the quiz.

Counting functions

General functions from a finite set $\mathbb{X}$ to another finite set $\mathbb{Y}$ can be counted too. It is easy to construct a function:

> For every x in the domain, choose one (and only one) y in the co-domain and include the pair (x, y) in function, that is $f(x) = y$.

For instance, you can construct 3^2 functions from $\{a, b\}$ to $\{0, 1, 2\}$: Map a to one of $0, 1, 2$ and map b to one of $0, 1, 2$. Each of these 9 functions can be represented by one of the 9 sets below.

$$\begin{array}{lll} f_0 = \{(a, 0), (b, 0)\} & f_1 = \{(a, 0), (b, 1)\} & f_2 = \{(a, 0), (b, 2)\} \\ f_3 = \{(a, 1), (b, 0)\} & f_4 = \{(a, 1), (b, 1)\} & f_5 = \{(a, 1), (b, 2)\} \\ f_6 = \{(a, 2), (b, 0)\} & f_7 = \{(a, 2), (b, 1)\} & f_8 = \{(a, 2), (b, 2)\} \end{array}$$

To change the example, if $|\mathbb{X}| = 6$ and $|\mathbb{Y}| = 5$, then 5^6 functions from $\mathbb{X}$ to $\mathbb{Y}$ can be defined. In general, if $|\mathbb{X}| = n$ and $|\mathbb{Y}| = m$, there are

$$m^n = |\mathbb{Y}|^{|\mathbb{X}|}$$

Using sets and subscripted functions symbols is just another way to *name* functions. Usually, people write

$$f : \mathbb{X} \to \mathbb{Y}$$

or

$$y = f(x)$$

to name functions. A description of f tells someone how to compute y given x.

For each of the six x's choose one of the five y's to map x to. That's 5 choices 6 times.

functions from $\mathbb{X}$ to $\mathbb{Y}$.

Another way to count functions is by counting *adjacency matrices* that represent functions. Let $f : \mathbb{X} \to \mathbb{Y}$ be a function, and let $|\mathbb{X}| = n$ and $|\mathbb{Y}| = m$. Then f is an $n \times m$ adjacency matrix.

For instances, to count functions from a 3-element set $\mathbb{X} = \{a, b, c\}$ into a 4-element $\mathbb{Y} = \{0, 1, 2, 3\}$, you must fill in a 3×4 adjacency matrix. For instance, the matrix

	0	1	2	3
a	0	0	1	0
b	0	1	0	0
c	0	0	1	0

defines a function $a \to 2$, $b \to 1$, $c \to 2$,

There are $3 \times 4 = 12$ entries in the matrix. For the matrix to represent a functions, there can be one-and-only one `True` value in each row. For each of the 3 rows, there are 4 choices of where to place the 1. Therefore, there are $4^3 = |\mathbb{Y}|^{|\mathbb{X}|}$ ways to fill the matrix. Therefore, there are 4^3 functions.

A functional adjacency matrix

		m columns					
		0	1	2	$\cdots$	$(m-2)$	$(m-1)$
	0						
n	1						
rows	2			Fill with 0's and 1's			
	$\vdots$			But only one 1 in each row			
	$n-1$						

Each row will have a 1 in one and only one column. There are m column to choose from for each of the n rows. Therefore, there are m^n functions from $\mathbb{X}$ to $\mathbb{Y}$.

As a "counting functions" example, let

$$\mathbb{X} = \{0, 1\} \quad \text{and} \quad \mathbb{Y} = \{a, b, c\}$$

There are $3^2 = 9$ functions from $\mathbb{X}$ to $\mathbb{Y}$.

1. $\begin{array}{c c} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} 0 \\ 1 \end{array} & \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \end{array}$

2. $\begin{array}{c c} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} 0 \\ 1 \end{array} & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \end{array}$

3. $\begin{array}{c c} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} 0 \\ 1 \end{array} & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$

4. $\begin{array}{c c} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} 0 \\ 1 \end{array} & \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \end{array}$

5. $\begin{array}{c c} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} 0 \\ 1 \end{array} & \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix} \end{array}$

6. $\begin{array}{c c} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} 0 \\ 1 \end{array} & \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$

7. $\begin{array}{c c} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} 0 \\ 1 \end{array} & \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \end{array}$

8. $\begin{array}{c c} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} 0 \\ 1 \end{array} & \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \end{array}$

9. $\begin{array}{c c} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} 0 \\ 1 \end{array} & \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$

Theorem 15: Counting functions

There are $|\mathbb{Y}|^{|\mathbb{X}|}$ functions from $\mathbb{X}$ to $\mathbb{Y}$.

Proof: Counting functions

A direct proof *is easiest here. For each $x \in \mathbb{X}$ there are $|\mathbb{Y}|$ choices of y to which x can be mapped.*
In total, there are $|\mathbb{Y}|^{|\mathbb{X}|}$ ways to construct a function.

The matrix below shows the count of functions as the cardinality of the domain $\mathbb{X}$ and range $\mathbb{Y}$ vary.

Count of functions

$|\mathbb{Y}|^{|\mathbb{X}|}$ functions from $\mathbb{X}$ to $\mathbb{Y}$

		$\lvert\mathbb{Y}\rvert$					
		0	1	0	3	4	5
	0	1	1	0	0	0	0
	1	0	1	2	3	4	5
	2	0	1	4	9	16	25
$\lvert\mathbb{X}\rvert$	3	0	1	8	27	64	125
	4	0	1	4	81	256	625
	5	0	1	32	243	1024	3125
	6	0	1	64	729	4096	15,625

Let's name the empty function ϕ.

$$\phi : \emptyset \to \mathbb{Y}$$

For technical reasons related to category theory, there must be a function from $\emptyset$ to itself. That's why there is a 1 in row 0, column 0 in array on the left

You can test if $f : \mathbb{X} \to \mathbb{Y}$ is a one-to-one or onto function by looking at f's adjacency matrix.

- A function f is onto if every column in its adjacency matrix has at least one 1. For instance, the function represented by the adjacency matrix

		$\mathbb{Y}$		
		a	b	c
	0	0	0	1
$\mathbb{X}$	1	1	0	0
	2	0	1	0
	3	0	1	0

  is onto: Each element a, b, c in $\mathbb{Y}$ is the image of some x in $\mathbb{X}$.

  There can be no onto functions from $\mathbb{Y}$ onto $\mathbb{X}$: Notice how $b \in \mathbb{Y}$ maps to two values: 2 and 3 in $\mathbb{X}$. This makes the "inverse" a relation, and keeps it from being a function. This is also an example of the pigeonhole principle: You can't place four pigeons in three holes without some hole containing more than one bird.

  If some column is not covered, the function is not onto. For instance,

Consider the adjacency matrix

		$\mathbb{Y}$		
		a	b	c
	0	0	0	1
$\mathbb{X}$	1	0	1	0
	2	0	1	0
	3	0	0	1

The element $a \in \mathbb{Y}$ is not the image of any element in $\mathbb{X}$: Its column is all 0's.

- A function f is one-to-one if no column in its adjacency matrix has more than one 1. For instance, consider the adjacency matrix

		$\mathbb{Y}$				
		a	b	c	d	e
	0	0	0	1	0	0
$\mathbb{X}$	1	1	0	0	0	0
	2	0	1	0	0	0
	3	0	0	0	0	1

It represents a one-to-one function: No letter in $\mathbb{Y}$ is the image of more than one number in $\mathbb{X}$.

The function is not onto: It can't be, because four numbers cannot (functionally) cover five letters.

Therefore, the "inverse" is a *partial* function: It is not defined for input e.

Counting permutations

On a word, a permutation is an *anagram*. For instance, some anagrams of "mathematics" are

- "asthmatic me"
- "hammiest cat"
- "mismatch tea"
- "sematic math"

where I've included a extra space for readability.

Definition 16: Permutation

A permutation *is a one-to-one function from a set onto itself.*

For instance, there are 2 permutations on $\mathbb{B}$

$$\langle 0, 1\rangle \qquad \text{and} \qquad \langle 1, 0\rangle$$

Using these permutations on $\mathbb{B}$, you can create 6 permutations on $\{0, 1, 2\}$.

- Insert 2 last, in the middle, and first in $\langle 0, 1\rangle$.

$$\langle 0, 1, 2\rangle, \quad \langle 0, 2, 1\rangle, \quad \langle 2, 0, 1\rangle$$

- Insert 2 last, in the middle, and first in $\langle 1, 0\rangle$.

$$\langle 1, 0, 2\rangle, \quad \langle 1, 2, 0\rangle, \quad \langle 2, 1, 0\rangle$$

In general, given a permutation on $n-1$ elements, say on $n-1=5$ letters

$$\langle a, b, c, d, e\rangle$$

You can create a permutation on n (in this case $n=6$) elements by inserting the new element f in one of $n=6$ positions, for instance

$\langle a, b, c, d, e\rangle \to \langle a, b, c, d, e, f\rangle$	insert last
$\langle a, b, c, d, e\rangle \to \langle a, b, c, d, f, e\rangle$	insert next to last
$\langle a, b, c, d, e\rangle \to \langle a, b, c, f, d, e\rangle$	insert fourth
$\langle a, b, c, d, e\rangle \to \langle a, b, f, c, d, e\rangle$	insert third
$\langle a, b, c, d, e\rangle \to \langle a, f, b, c, d, e\rangle$	insert second
$\langle a, b, c, d, e\rangle \to \langle f, a, b, c, d, e\rangle$	insert first

This leads to recurrence equation/initial condition

$$\pi_n = n\pi_{n-1}, \qquad \pi_0 = 1$$

where π_n is the number of permutations on n things. The solution to the recurrence is

$$\pi(n) = \pi_n = n! \qquad \text{"called } n \text{ factorial "}$$

! is the Unix bang command. It is a useful shorthand when typing shell commands. Does anyone do that anymore?

Theorem 16: Counting permutations

There are $|\mathbb{X}|\,! = n!$ permutations on a set $\mathbb{X}$ with cardinality $|\mathbb{X}| = n$.

A new object can be inserted in a permutation of n objects in $n+1$ places.

$$\begin{array}{ccccccccc} \downarrow & & \downarrow & & \downarrow & & \cdots & & \downarrow \\ & x_1 & & x_2 & & x_2 & \cdots & x_n & \end{array}$$

Proof: Counting permutations

We have seen the theorem is `True` *for small cases, establishing a* basis *for induction.*
Assume *the theorem in* `True` *for some* n. Then *given one of the* $n!$ *permutations on* n *objects, you can insert a new* $(n+1)^{st}$ *object inserted into the given permutation in* $n+1$ *ways. This generated* $n+1$ *permutations for each of the* $n!$ *original permutations. Thus, there are* $(n+1)\cdot n! = (n+1)!$ *permutations on* $(n+1)$ *objects.*

Pass a Quiz: Counting functions and permutations

Take a quiz on page 326 to check your understanding. You can return to here from the quiz.

Counting relations

Relations among people are common: Father-Son, Mother-Daughter, Brother-Sister, Myself-Friend; but complicated.

A relation is a generalization of function.

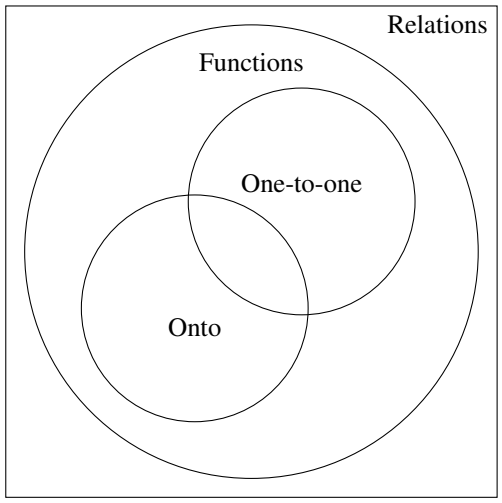
- A function is *deterministic*: Given input x the output y is completely determined. For instance, the value of y is completely determined by the function

$$y = f(x) = x^2 - x - 1$$

- A relation can be *non-deterministic*: Given input x there may be many possible output values. For instance, the value of y is not completely determined by the relation

$$y \leq x \quad \text{many values of } y \text{ satisfy the inequality.}$$

The Venn diagram below shows associations between relations and functions.



Definition 17: Relation

A relation from $\mathbb{X}$ to $\mathbb{Y}$ is a subset of the Cartesian product

$$\mathbb{X} \times \mathbb{Y} = \{(x, y) : x \in \mathbb{X} \wedge y \in \mathbb{Y}\}$$

There are several important relations. Some are well-known:

- Equality: $a = b$, if $a - b = 0$.
- Less than: $a < b$, if $a - b < 0$
- Greater than $a > b$ if $a - b > 0$.
- Not equal: $a \neq b$, if $a - b \neq 0$.

There are many important relations that are not be as well-known. Here are two of that are useful because they demonstrate general concepts.

Divides: 4 divides 28 since $28 = 4 \cdot 7$
Congruence: $3 \equiv 15 \pmod 4$ since 4 divides $3 - 15 = -12$.

- Divides: $a \mid b$, if $b = ac$ for some natural number c.
- Congruence: $a \equiv b \bmod n$, if $n \mid (b - a)$.

One way to count relations is to recognize a relation can be represented as an adjacency matrix. Let $\sim$ be a relation from $\mathbb{X}$ to $\mathbb{Y}$. Let $|\mathbb{X}| = n$ and $|\mathbb{Y}| = m$. Label the elements in $\mathbb{X}$ by $0, 1, \ldots, (n-1)$. Label the elements in $\mathbb{Y}$ by $0, 1, \ldots, (m-1)$. Then the relation $\sim$ can be expressed as an $n \times m$

adjacency matrix.

A relational adjacency matrix

		m columns					
		0	1	2	$\cdots$	$(m-2)$	$(m-1)$
	0						
	1						
n	2			Fill with 0's and 1's			
rows	$\vdots$			in any way you like			
	$(n-2)$						
	$(n-1)$						

There are nm entries in the adjacency matrix. Each of the nm entries will be either `True` (1) or `False` (0).

- The value in row i, column j is 0 if $i \nsim j$ (`False` if n is not related to m)

- The value in row i, column j is 1 if $i \sim j$ (`True` if n is related to m)

There are 2^{nm} ways to fill-out an adjacency matrix matrix. Therefore, there are 2^{nm} relations between $\mathbb{X}$ and $\mathbb{Y}$.

For example, there are $16 = 2^{2\cdot 2}$ relations on the bits $\mathbb{B} = \{0, 1\}$. You can

write them down.

The empty relation

	0	1
0	0	0
1	0	0

$0 \sim 0$

	0	1
0	1	0
1	0	0

$a < b$

	0	1
0	0	1
1	0	0

$a > b$

	0	1
0	0	0
1	1	0

$1 \sim 1$

	0	1
0	0	0
1	0	1

Equality

	0	1
0	1	0
1	0	1

Not Equal

	0	1
0	0	1
1	1	0

$(0 \sim 0) \wedge (0 \sim 1)$

	0	1
0	1	1
1	0	0

$(1 \sim 0) \wedge (1 \sim 1)$

	0	1
0	0	0
1	1	1

$(0 \sim 0) \wedge (1 \sim 0)$

	0	1
0	1	0
1	1	0

$(0 \sim 1) \wedge (1 \sim 1)$

	0	1
0	0	1
1	0	1

$1 \nsim 1$

	0	1
0	1	1
1	1	0

$0 \nsim 0$

	0	1
0	0	1
1	1	1

$a \leq b$

	0	1
0	1	1
1	0	1

$a \geq b$

	0	1
0	1	0
1	1	1

Appalachian relation

	0	1
0	1	1
1	1	1

I apologize if you find my use of the name "Appalachian" insulting. My intent is solely fix these ideas in your mind.

Theorem 17: Counting relations

Let $|\mathbb{X}| = n$ and $|\mathbb{Y}| = m$. There are

$$2^{nm} = 2^{|\mathbb{X}| \times |\mathbb{Y}|} \quad \text{relations from } \mathbb{X} \text{ to } \mathbb{Y}.$$

The matrix below shows the count of relations as the cardinalities of the

domain $\mathbb{X}$ and range $\mathbb{Y}$ vary.

Count of relations

2^{nm} relations from $\mathbb{X}$ to $\mathbb{Y}$

		$m = \lvert\mathbb{Y}\rvert$					
		0	1	2	3	4	5
	0	1	1	1	1	1	1
	1	1	2	4	8	16	32
	2	1	4	16	64	256	1024
$n = \lvert\mathbb{X}\rvert$	3	1	8	64	512	4096	32,768
	4	1	16	256	4096	65,536	1,048,576
	5	1	32	1024	32,768	1,048,576	33,554,432
	6	1	64	4096	262,144	16,777,216	1,073,741,824

Pass a Quiz: Counting relations

Take a quiz on page 327 to check your understanding. You can return to here from the quiz.

Counting subsets

Given a set you can form subsets. For instance, given the set of bits $\mathbb{B} = \{0, 1\}$, you can form $2^2 = 4$ subsets

$$\emptyset, \{0\}, \{1\}, \{0, 1\}$$

Recall the definition from the notes on sets.

Definition 18: Subset

Let $\mathbb{X}$ and $\mathbb{Y}$ be sets. $\mathbb{X}$ is a subset *of $\mathbb{Y}$, written*

$$\mathbb{X} \subseteq \mathbb{Y}$$

if every element in $\mathbb{X}$ is also an element of $\mathbb{Y}$.

$$(a \in \mathbb{X}) \Rightarrow (a \in \mathbb{Y})$$

A *proper subset* is a subset that is not all of the enclosing set.

The empty set $\emptyset$. Some people use $\{\}$ to write $\emptyset$. The empty set $\emptyset$ is the set without any elements. The empty set $\emptyset$ is a subset of every set.

It is interesting to count the number of subsets of a given size. Using the diagram to the left,

Binomial Coefficients $\binom{n}{k}$

		Choose k			
		0	1	2	3
	0	1			
n	1	1	1		
	2	1	2	1	
	3	1	3	3	1

Subset $\subseteq$ and *proper subset* $\subset$ are similar to *less than or equal* $\leq$ and *less than* $<$. An example of *round* versus *pointy* notation to describe similar ideas in different contexts.

Definition 19: Proper subset

Let $\mathbb{X}$ and $\mathbb{Y}$ be sets. $\mathbb{X}$ is a proper subset *of $\mathbb{Y}$, written*

$$\mathbb{X} \subset \mathbb{Y}$$

if $\mathbb{X}$ is a subset of $\mathbb{Y}$, but $\mathbb{X}$ is not equal to $\mathbb{Y}$.

$$(\mathbb{X} \subseteq \mathbb{Y}) \wedge (\mathbb{X} \neq \mathbb{Y})$$

Given a set $\mathbb{X}$ with n there are two basic problems related to subsets:

1. How many subsets does $\mathbb{X}$ have? Can you express it as a function?
2. How many subsets of a given size (cardinality) does $\mathbb{X}$ have? Can you express it as a function?

To solve these problems it is useful to list the subsets of small sets and arrange them by the number of elements they contain.

Subsets of $\emptyset$

No elements
$\emptyset$

Subsets of $\{0\}$

No elements	1 element
$\emptyset$	$\{0\}$

Subsets of $\{0, 1\}$

No elements	1 element	2 elements
$\emptyset$	$\{0\}$	$\{0, 1\}$
	$\{1\}$	

Subsets of $\{0, 1, 2\}$

No elements	1 element	2 elements	3 elements
$\emptyset$	$\{0\}$	$\{0, 1\}$	$\{0, 1, 2\}$
	$\{1\}$	$\{0, 2\}$	
	$\{2\}$	$\{1, 2\}$	

Look at how you can use the subsets of $\{0, 1, 2\}$ above to create subsets of $\{0, 1, 2, 3\}$

First, all subsets of $\{0, 1, 2\}$ are subsets of $\{0, 1, 2, 3\}$

Second, if $\mathbb{X}$ is a $k-1$ element subset of $\{0, 1, 2\}$, then $\mathbb{X} \cup \{3\}$ is a k element subset of $\{0, 1, 2, 3\}$,

This is demonstrated in the diagram below, which expands the $8 = 2^3$ subsets of $\{0, 1, 2\}$ to 16 subsets of $\{0, 1, 2, 3\}$.

Do you see the color coding of subsets?

Subsets of $\{0, 1, 2, 3\}$

No elements	1 element	2 elements	3 elements	4 elements
$\emptyset$	$\{0\}$	$\{0, 1\}$	$\{0, 1, 2\}$	$\{0, 1, 2, 3\}$
	$\{1\}$	$\{0, 2\}$	$\{0, 1, 3\}$	
	$\{2\}$	$\{1, 2\}$	$\{0, 2, 3\}$	
	$\{3\}$	$\{0, 3\}$	$\{1, 2, 3\}$	
		$\{1, 3\}$		
		$\{2, 3\}$		

There are several things you can notice about constructing subsets. But first, we should name things.

Definition 20: Binomial coefficient

The symbol

$$\binom{n}{k}$$

stands for the count of k-element subsets of an n-element set. $\binom{n}{k}$ is called a binomial coefficient.

$\binom{n}{k}$ is a binomial coefficient and is called "n choose k." These *boundary conditions* hold:

Other notations for binomial coefficients include

$$C(n, k), \quad {}_nC_k$$

where C stands for *combination*.

- The $\emptyset$ is the only subset of $\mathbb{X}$ with $k = 0$ elements. Therefore

$$\binom{n}{0} = 1 \quad \text{for all } n \in \mathbb{N}, \quad \text{there is only 1 way to choose nothing.}$$

- $\mathbb{X}$ is the only subset of $\mathbb{X}$ with $k = n$ elements. Therefore

$$\binom{n}{n} = 1 \quad \text{for all } n \in \mathbb{N}, \quad \text{there is only 1 way to choose everything.}$$

- If $k > n$, you cannot choose k elements from fewer (n) elements. Therefore

$$\binom{n}{k} = 0 \quad \text{for all } k > n, \quad \text{you can't choose more than you have.}$$

There are some other facts about binomial coefficients that you can deduce.

A *singleton* is set with only 1 member.

- There are n singleton subsets of $\mathbb{X}$. Therefore

$$\binom{n}{1} = n \quad \text{for all } n \geq 1.$$

- Choosing k elements to put in a subset is equivalent to choosing $n-k$ elements to *not* put in the subset. Therefore

$$\binom{n}{k} = \binom{n}{n-k} \quad \text{for all } 0 \leq k \leq n,\ n \in \mathbb{N}.$$

Perhaps you noticed that the number of subsets doubles each time the cardinality of $\mathbb{X}$ is increased by 1. This idea can be used to answer the question: How many subsets does $\mathbb{X}$ have? And, express the answer as a function?

Theorem 18: Counting all subsets

There are 2^n subsets of an n-element set.

Proof: Counting subsets

There is 1 subset of the empty set, it is a subset of itself: $\emptyset \subseteq \emptyset$. By the above reasoning, the number of subsets doubles each time another element is put in the starting set. That is, when the size of the set, its cardinality, increases from $n-1$ to n, the number of subsets doubles: All the old subsets are still subsets. And, you can insert the new element in each of these subsets creating just as many new subsets. Bluntly, let d_n be the number of subsets of an n-element set. Then d_n satisfies the Doubling recurrence equation

$$d_n = 2d_{n-1},\ d_0 = 1$$

Terms d_n in the Doubling sequence can be computed by the Doubling function

$$d(n) = 2^n$$

A corollary is: The sum of counts of subsets of the various sizes $k = 0, 1, \ldots, n$ is 2^n must be 2^n. The sum of binomial coefficients $\binom{n}{k}$ for $k = 0, 1, \ldots, n$ is 2^n.

Setting $x = y = 1$ in the Binomial theorem

$$(x+y)^n = \binom{n}{0}x^n + \binom{n}{1}x^{n-1}y + \cdots + \binom{n}{n}y^n = \sum_{k=0}^{n}\binom{n}{k}x^{n-k}y^k$$

provides another proof of the corollary.

Corollary 1: Pascal's triangle: Row sum is 2^n

Let $\mathbb{X}$ have cardinality n. There are $\binom{n}{k}$ (say n choose k) k-element subsets of $\mathbb{X}$. The sum of these binomial coefficients for each k from

0 up to $k = n$, is the total number of subsets 2^n of $\mathbb{X}$. That is,

$$\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{n} = \sum_{k=0}^{n} \binom{n}{k} = 2^n$$

Counting of subsets by their size

Consider counting the number of ways you can choose k things from a set with n things. That is, computing the value of the binomial coefficient $\binom{n}{k}$.

Look again at the $(n-1) = 4$ element set $\mathbb{X} = \{0, 1, 2, 3\}$ There are 4 subsets of $\mathbb{X}$ with cardinality 3. There are 6 subsets of $\mathbb{X}$ with cardinality 2. Using these subsets you can create 10 three-element subsets of $\mathbb{X} \cup \{4\} = \{0, 1, 2, 3, 4\}$.

Copy 3-element subsets of $\mathbb{X} = \{0, 1, 2, 3\}$ as 3-element subsets of $\mathbb{X} \cup \{4\} = \{0, 1, 2, 3, 4\}$. Insert 4 into each 2-element subset of $\mathbb{X}$.

$$\begin{aligned}
\{0, 1, 3\} &\to \{0, 1, 3\} \\
\{0, 2, 3\} &\to \{0, 2, 3\} \\
\{1, 2, 3\} &\to \{1, 2, 3\} \\
\{0, 1, 2\} &\to \{0, 1, 2\} \\
\{0, 3\} &\to \{0, 3, 4\} \\
\{1, 3\} &\to \{1, 3, 4\} \\
\{2, 3\} &\to \{2, 3, 4\} \\
\{0, 1\} &\to \{0, 1, 4\} \\
\{0, 2\} &\to \{0, 2, 4\} \\
\{0, 3\} &\to \{0, 3, 4\}
\end{aligned}$$

The numbers generated in this fashion fill Pascal's triangle. The value in row n and column k is $\binom{n}{k}$, the number of k-element subsets of an n-element set.

Pascal's triangle

Binomial Coefficients $\binom{n}{k}$

		Choose k										
		0	1	2	3	4	5	6	7	8	9	
	0	1										
	1	1	1									
	2	1	2	1								
	3	1	3	3	1							
n	4	1	4	6	4	1						
	5	1	5	10	10	5	1					
	6	1	6	15	20	15	6	1				
	7	1	7	21	35	35	21	7	1			
	8	1	8	28	56	70	56	28	8	1		
	9	1	9	36	84	126	126	84	36	9	1	
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

The value in row n, column k of Pascal's triangle is the number of ways k elements can be chosen from a set of n elements. The notation for the value in row n, column k is

$$\binom{n}{k} \quad \text{and is called "}n\text{ choose }k\text{."}$$

For instance, $\binom{6}{4} = 15$ states there are 15 ways to choose 4 things from a set of 6 things.

And $\binom{8}{2} = 28$ states there are 28 ways to choose 2 things from 8.

The values in Pascal's triangle can be computed a two-dimensional recursion equation known as Pascal's identity or Pascal's rule.

Theorem 19: Pascal's identity

Binomial coefficients satisfy the recurrence known as Pascal's identity

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

with *boundary conditions*

$$\binom{n}{0} = 1, \ (n = 0, 1, \ldots) \qquad \binom{n}{n} = 1, \ (n = 1, 2, \ldots)$$

Proof: Pascal's identity

A direct proof *goes like this.*
Given a set $\mathbb{X}$ with $n-1$ elements. Let

$$\mathbb{X}' = \mathbb{X} \cup \{y\}, \quad \text{where } y \notin \mathbb{X}.$$

$\{X\}'$ has n elements, its cardinality is n.
By notation, there are $\binom{n-1}{k}$ subsets of $\mathbb{X}$ with k elements. Each of these is also a subset of $\mathbb{X}'$.
And, there are $\binom{n-1}{k-1}$ subsets of $\mathbb{X}$ with $k-1$ elements. The set $\{y\}$ can be inserted (unioned) into each of these to create a k element subset of $\mathbb{X}'$.
Therefore, the number of k-element subsets of $\mathbb{X}'$ is the number of k-element subsets of $\mathbb{X}$ plus the number of $k-1$-element subsets of $\mathbb{X}$.

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

Notice when you fill Pascal's triangle with 0's to form a rectangular array: The first column is the Alice sequence; the second column is the Gauss sequence; the third column is the triangular sequence; numbers in the fourth column are called *pyramidal numbers*, and so on into higher dimensional spaces.

The column labeled 0 is the first column, the column labeled 1 is the second column, the column labeled 2 is the third column, and so on.

Pascal's rectangle

Binomial Coefficients $\binom{n}{m}$

		Choose m									
		0	1	2	3	4	5	6	7	8	9
	0	1	0	0	0	0	0	0	0	0	0
	1	1	1	0	0	0	0	0	0	0	0
	2	1	2	1	0	0	0	0	0	0	0
	3	1	3	3	1	0	0	0	0	0	0
n	4	1	4	6	4	1	0	0	0	0	0
	5	1	5	10	10	5	1	0	0	0	0
	6	1	6	15	20	15	6	1	0	0	0
	7	1	7	21	35	35	21	7	1	0	0
	8	1	8	28	56	70	56	28	8	1	0
	9	1	9	36	84	126	126	84	36	9	1
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Naming values in Pascal's triangle by binomial coefficient notation is impor-

tant. A side-by-side view can help you remember this.

Values of binomial coefficients

	0	1	2	3	4	5	0	1	2	3	4	5
0	1						$\binom{0}{0}$					
1	1	1					$\binom{1}{0}$	$\binom{1}{1}$				
2	1	2	1				$\binom{2}{0}$	$\binom{2}{1}$	$\binom{2}{2}$			
3	1	3	3	1			$\binom{3}{0}$	$\binom{3}{1}$	$\binom{3}{2}$	$\binom{3}{3}$		
4	1	4	6	4	1		$\binom{4}{0}$	$\binom{4}{1}$	$\binom{4}{2}$	$\binom{4}{3}$	$\binom{4}{4}$	
5	1	5	10	10	5	1	$\binom{5}{0}$	$\binom{5}{1}$	$\binom{5}{2}$	$\binom{5}{3}$	$\binom{5}{4}$	$\binom{5}{5}$

To count the number of k-element subsets, let's consider one way you how you could form a k-element subset.

1. Line each of the n elements up in a list: From your knowledge of permutations, you know there are $n!$ ways to do this.

   The chart below shows the first element can be chosen in n ways; then the second element can be chosen in $n-1$ ways; and the third in $n-2$ ways; down to 2 and 1 ways to choose the penultimate and ultimate elements.

   $$\begin{array}{ccccccccccc} n & \times & (n-1) & \times & (n-2) & \times & \cdots & \times & 2 & \times & 1 \\ \text{first} & & \text{second} & & \text{third} & & \cdots & & (n-1)^{\text{st}} & & n^{\text{th}} \end{array}$$

   This first step in counting k-element subsets of an n-element set maps n to $n!$

   $$n \longrightarrow n!$$

2. Now throw away (divide out) the last $(n-k)$ elements in the list, leaving only the first k elements: The last $(n-k)$ elements can be arranged in $(n-k)!$ ways. This reduces our count of subsets from $n!$ to $n!/(n-k)!$.

   $$n! \longrightarrow \frac{n!}{(n-k)!}$$

   .

3. Recognize that the k remaining chosen elements can be arranged in $k!$ ways, and this is immaterial with respect to sets. Our count of subsets is reduced to

   $$\boxed{\frac{n!}{(n-k)!} \longrightarrow \frac{n!}{k!(n-k)!} = \binom{n}{k}.}$$

This is the functional form for binomial coefficients

Theorem 20: Binomial coefficients: Factorial form

There are

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

k-element subsets of an n-element set.

Using the theorem you can write the recurrence relation

$$\binom{n}{k} = \frac{n}{k}\binom{n-1}{k-1}$$

which leads to an efficient recursive algorithm to compute n choose k.

$$\begin{aligned}\binom{n}{k} &= \frac{n!}{k!(n-k)!}\\ &= \frac{n}{k}\frac{(n-1)!}{(k-1)!(n-k)!}\\ &= \frac{n}{k}\binom{n-1}{k-1}\end{aligned}$$

```
choose :: Int -> Int -> Int
choose n 0 = 1
choose 0 k = 0
choose n+1 k+1 = (choose n k) * (n+1) `div` (k+1)
```

Power set of a set

The collection of all subsets of a set $\mathbb{Y}$ is called the power set of $\mathbb{Y}$.

Definition 21: Power set of a set

Let $\mathbb{Y}$ be a set. The power set of $\mathbb{Y}$, written $2^{\mathbb{Y}}$, is the set of all subsets of $\mathbb{Y}$. That is,

$$2^{\mathbb{Y}} = \{\mathbb{X} : \mathbb{X} \subseteq \mathbb{Y}\}$$

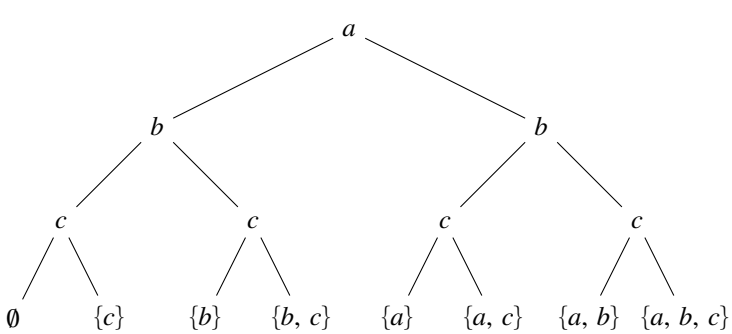
The power set of $\mathbb{A}$ is denoted by $2^{\mathbb{A}}$.

If $\mathbb{A}$ has 3 elements then $\mathbb{A}$'s power set contains $2^3 = 8$ elements, each of which is a subset of $\mathbb{A}$.

Think about how you might construct all subsets of $\mathbb{X}$. A binary decision tree provides a simple algorithm.

For each element x make one of two choices: include x in the subset or leave it out.

Consider the decision tree for constructing subsets of $\mathbb{X} = \{a, b, c\}$. When the left branch is followed, the element is not included in the subset. When the right branch is followed, the element is included in the subset.



Theorem 21: Cardinality of a power set

A set $\mathbb{X}$ with n elements has 2^n subsets. That is, the power set $2^{\mathbb{X}}$ of $\mathbb{X}$ contains all 2^n subsets of $\mathbb{X}$.

The cardinality of the empty set $\emptyset$ is $|\emptyset| = 0$.

Proof: Cardinality of a power set

An inductive proof goes like this. First, there is a basis*:*

- *The power set of the empty set $\emptyset$ is $\{\emptyset\}$, which has cardinality $1 = 2^0$.*
- *The power set of a singleton $\{a\}$ is $\{\emptyset, \{a\}\}$, which has cardinality $2 = 2^1$.*
- *And, there are $4 = 2^2$ and $8 = 2^3$ subsets of a 2-element and 3-element, respectively.*

Next, there is a hypothesis*: For some n, any n-element set $\mathbb{X}$ has 2^n subsets. Finally, there is a* conclusion*: If $\mathbb{X}' = \mathbb{X} \cup \{y\}$ for some $y \notin \mathbb{X}$, then $\mathbb{X}'$ has $2^{(n+1)}$ subsets: 2^n subsets that do not include y and 2^n subsets that do.*

If $|\mathbb{X}| = n$, then the cardinality of the power set of $\mathbb{X}$ is $\left|2^{\mathbb{X}}\right| = 2^{|\mathbb{X}|} = 2^n$.

Pass a Quiz: Counting subsets

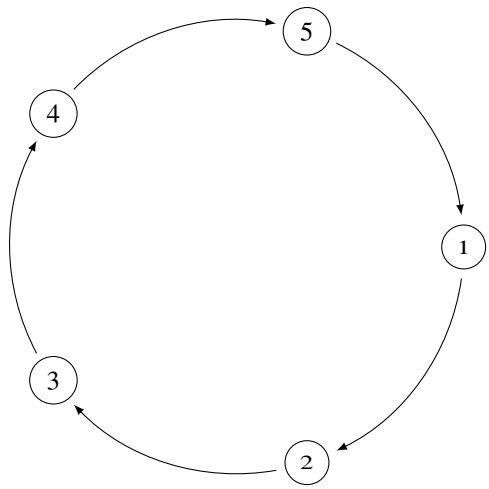
Take a quiz on page 326 to check your understanding. You can return to here from the quiz.

Counting permutations by cycles

Stirling's numbers of the first kind count the number of permutations with a given number of cycles. This will be explained in more detail in the notes on sorting data & permutations. But, for now, let's just derive the recurrence that counts permutations with a given number of cycles.

$\left[{n \atop k}\right]$ is read "n cycle k."

The Stirling's number of the first kind $\left[{n \atop k}\right]$ counts the number of ways to arrange n objects into k cycles. A cycle is like placing beads on a necklace. The clockwise cycle



can be written using *cycle* notation as

$$[1, 2, 3, 4, 5] = [2, 3, 4, 5, 1] = [3, 4, 5, 1, 2] = [4, 5, 1, 2, 3] = [5, 1, 2, 3, 4]$$

This permutation describes the mapping from one sequence to another. Apply this permutation a series of times to the normal order. You'll see the numbers shift right and cycle from the last position to the first position.

$$\begin{aligned} \langle 1, 2, 3, 4, 5\rangle &\to \langle 5, 1, 2, 3, 4\rangle && \text{apply once} \\ &\to \langle 4, 5, 1, 2, 3\rangle && \text{apply twice} \\ &\to \langle 3, 4, 5, 1, 2\rangle && \text{apply thrice} \end{aligned}$$

The order in of numbers in the cycle is important. For instance,

$$[1, 5, 4, 3, 2] \quad \text{describes a counterclockwise rotation.}$$

And,

$$[1, 2, 3, 4, 5] \neq [2, 1, 3, 4, 5]$$

because you cannot interchange the 1 and 2 without breaking the necklace.

A permutation can have more than one cycle.

$$[1, 2, 3][4, 5]$$

is a two-cycle permutation on the five numbers. Apply this permutation a series of times to the normal order. You'll see the first three and last two numbers cycle among themselves.

$$\begin{aligned} \langle 1, 2, 3, 4, 5\rangle &\to \langle 3, 1, 2, 5, 4\rangle && \text{apply once} \\ &\to \langle 2, 3, 1, 4, 5\rangle && \text{apply twice} \\ &\to \langle 1, 2, 3, 5, 4\rangle && \text{apply thrice} \end{aligned}$$

It can be shown that there are 50 different two cycles on 5 elements, but that is not obvious.

What may be obvious is:

- A non-empty set cannot be represented without any cycles at all: $\begin{bmatrix} n \\ 0 \end{bmatrix} = 0$.
- And, n elements can be written in n cycles in only 1 way: $\begin{bmatrix} n \\ n \end{bmatrix} = 1$.

These are the *boundary conditions* for Stirling's numbers of the first kind.

$$\begin{bmatrix} n \\ 0 \end{bmatrix} = 0 \quad \text{for } n > 0 \quad \text{and} \quad \begin{bmatrix} n \\ n \end{bmatrix} = 1 \quad \text{for all } n \tag{11}$$

Do you see the factorial sequence in the column labeled 1?

$$\begin{bmatrix} n \\ 1 \end{bmatrix} = (n-1)!$$

you see the triangular sequence in diagonal?

$$\begin{bmatrix} n \\ n-1 \end{bmatrix} = \binom{n}{2} = \frac{n(n-1)}{2}$$

Stirling's first rectangle

Stirling Numbers of the First Kind $\begin{bmatrix} n \\ k \end{bmatrix}$

		Cycle k									
		0	1	2	3	4	5	6	7	8	9
	0	1	0	0	0	0	0	0	0	0	0
	1	0	1	0	0	0	0	0	0	0	0
	2	0	1	1	0	0	0	0	0	0	0
	3	0	2	3	1	0	0	0	0	0	0
n	4	0	6	11	6	1	0	0	0	0	0
	5	0	24	50	35	10	1	0	0	0	0
	6	0	120	274	225	85	15	1	0	0	0
	7	0	720	1764	1624	735	175	21	1	0	0
	8	0	5040	13068	13132	6769	1960	322	28	1	0
	9	0	40320	109584	118124	67284	22449	4336	546	36	1

Theorem 22: Stirling's identity of the first kind

Stirling's numbers of the first kind can be generated by the two-dimensional recurrence equation

$$\begin{bmatrix} n \\ k \end{bmatrix} = (n-1)\begin{bmatrix} n-1 \\ k \end{bmatrix} + \begin{bmatrix} n-1 \\ k-1 \end{bmatrix}$$

together with the *boundary conditions*

$$\begin{bmatrix} n \\ 0 \end{bmatrix} = 0, \ (n = 1, \ldots) \qquad \begin{bmatrix} n \\ n \end{bmatrix} = 1, \ (n = 0, 1, 2, \ldots)$$

The only functional form for Stirling's numbers of the first kind that I can point to is quite complex. I won't even try to write it down.

Counting partitions by subsets

Stirling's numbers of the second kind count partitions of a set. This will be explained in more detail in the notes on equivalences: how to partition a set. But, for now, let's just derive the recurrence that counts the number of partitions of a set into a given number of subsets.

Stirling's numbers of the second kind $\left\{ {n \atop k} \right\}$ counts the number of ways to partition a set into k subsets. For instance, consider the set $\mathbb{X} = \{1, 2, 3, 4, 5\}$. It is a partition of itself, and $\left\{ {5 \atop 1} \right\} = 1$. $\mathbb{X}$ can also be partitioned into three subsets: Some ways to do this are:

$$\{\{1, 2, 3\}, \{4\}, \{5\}\}, \quad \{\{1, 2\}, \{3, 4\}, \{5\}\}, \quad \{\{1\}, \{2, 4\}, \{3, 5\}\}$$

There are $\left\{ {5 \atop 3} \right\} = 25$ different ways to partition $\mathbb{X}$ into 3 subsets, but that is not obvious.

What may be obvious is:

- A non-empty set cannot be partitioned without any subsets at all: $\left\{ {n \atop 0} \right\} = 0$.
- And, an n-element set can be partitioned into n subsets in only 1 way: $\left\{ {n \atop n} \right\} = 1$.

Do you see the triangular sequence in the column labeled 2?

$$\left\{ {n \atop 3} \right\} = t(n) = \frac{n(n-1)}{2}$$

Do you see the triangular sequence in sub-diagonal?

$$\left\{ {n \atop n-1} \right\} = \binom{n}{2} = \frac{n(n-1)}{2}$$

Stirling's second rectangle

Stirling Numbers of the Second Kind $\left\{ {n \atop k} \right\}$

		Subset k									
		0	1	2	3	4	5	6	7	8	9
	0	1	0	0	0	0	0	0	0	0	0
	1	0	1	0	0	0	0	0	0	0	0
	2	0	1	1	0	0	0	0	0	0	0
	3	0	1	3	1	0	0	0	0	0	0
n	4	0	1	7	6	1	0	0	0	0	0
	5	0	1	15	25	10	1	0	0	0	0
	6	0	1	31	90	65	15	1	0	0	0
	7	0	1	63	301	350	140	21	1	0	0
	8	0	1	127	966	1701	1050	266	28	1	0
	9	0	1	255	3025	7770	6951	2646	426	36	1

Theorem 23: Stirling's identity of the second kind

Stirling's numbers of the second kind can be generated by the two-dimensional recurrence equation

$$\left\{ {n \atop k} \right\} = k\left\{ {n-1 \atop k} \right\} + \left\{ {n-1 \atop k-1} \right\}$$

together with the *boundary conditions*

$$\left\{ {n \atop 0} \right\} = 0,\ (n = 1, \ldots) \qquad \left\{ {n \atop n} \right\} = 1,\ (n = 0,\ 1,\ 2, \ldots)$$

together with the boundary conditions

$$\left\{ {n \atop 0} \right\} = 0 \quad \text{for } n > 0 \text{ and} \quad \left\{ {n \atop n} \right\} = 1 \quad \text{for all } n \tag{12}$$

The functional form for Stirling's numbers of the second kind is not so simple.

$$\left\{ {n \atop k} \right\} = \frac{1}{k!}\sum_{j=0}^{k}(-1)^{k-j}\binom{k}{j}j^n.$$

Do you see Mersenne sequence in the second column?

$$\left\{ {n \atop 2} \right\} = 2^{(n-1)}$$

Do you see the Triangular sequence in sub-diagonal?

$$\left\{ {n \atop n-1} \right\} = \binom{n}{2} = \frac{n(n-1)}{2}$$

Pass a Quiz: Stirling numbers

Take a quiz on page 327 to check your understanding. You can return to here from the quiz.

Homework Questions

Use your time outside of class to solve these problems.

1 How many truth assignments are there are 1, 2 and 3 Boolean variables? List the truth assignments. What is the pattern?

2 How many functions can be defined on the truth assignments in problem 1? What is the pattern?

3 Physicist estimate there are about 10^{80} atoms in the universe. How many inputs must a Boolean function have to be able to construct more functions than hydrogen atoms?

4 Let $\mathbb{B}^2 = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$ be the set of all Boolean pairs, and let $\mathbb{B}^3$ the the set of all Boolean triples.

1. How many functions can be defined from $\mathbb{B}^2$ to $\mathbb{B}^3$?

2. How many relations can be defined between $\mathbb{B}^2$ and $\mathbb{B}^3$?

5 Dr. S. Cooper assigned his assistant Ms. Jensen the task of listing all relations from $\mathbb{H}$ to $\mathbb{H}$, where $\mathbb{H}$ is the set of hexadecimal numerals. She politely refused. Why?

6 "Well then," Dr. Cooper said, "give me all functions from $\mathbb{D}$ to $\mathbb{H}$!" To which Ms. Jensen said, "Well maybe I can do that. Let me calculate." Why did she agree to consider this task?

7 The formula for counting functions is $|\mathbb{Y}|^{|\mathbb{X}|}$. When $\mathbb{Y} = \emptyset$ the formula states there are no functions from a non-empty set $\mathbb{X}$ to $\emptyset$. Does this jibe with the definition of a function given in the notes programming: with functions

$$(\forall x \in \mathbb{X})(\forall y_0, y_1 \in \mathbb{Y})((f(x) = y_0) \wedge (f(x) = y_1) \Rightarrow (y_0 = y_1))$$

That is, if $\mathbb{Y} = \emptyset$, is it `True` that there is no function f that maps each $x \in \mathbb{X}$ to at most one element in $\mathbb{Y} = \emptyset$.

8 In how many ways can you choose and permute n elements from $\mathbb{X}$ in the following instances

8.1 $n = 5$ and $\mathbb{X} = \mathbb{H}$.

8.2 $n = 5$ and $\mathbb{X} = \mathbb{E}$, the lower case English letters.

8.3 For arbitrary $n = 5$ and $\mathbb{X}$.

9 An Italian meal can contain up to 9 contains. They are listed below with the number of choices on a particular menu.

1. Aperitivo (5 choices)
2. Antipasto (4 choices)
3. Primo (2 choices)
4. Secondo (3 choices)
5. Contorno (5 choices)
6. Formaggio e frutta (1 choice)
7. Dolce (6 choice)
8. Caffè (4 choice)
9. Digestivo (5 choice)

9.1 If you are served one items from course, how many different meals could you have?

9.2 If you can decline a course, but still have no more than one item per course, how many different meals could you have?

9.3 If you on a date, how many meals can you and your date have?

10 A permutation is a one-to-one function from a finite set onto itself. Let $\vec{\pi} = \langle p_0, p_1, \ldots, p_{n-1} \rangle$ be a permutation of 0 to $n - 1$.

An *ascent* in a permutation is a pair $\left\langle p_{j-1}, p_j \right\rangle$ where $p_{j-1} < p_j$. The pair is a *descent* if $p_{j-1} > p_j$. For instance,

Ascents go up $\langle 5, 7 \rangle$ ascends: Descents go down $\langle 5, 3 \rangle$ descends.

- $\langle 0, 1, 2 \rangle$ has 2 ascents: $\langle 0, 1 \rangle$ and $\langle 1, 2 \rangle$
- $\langle 2, 1, 0 \rangle$ has no ascents,
- Each of $\langle 1, 0, 2 \rangle$, $\langle 0, 2, 1 \rangle$, $\langle 2, 0, 1 \rangle$, and $\langle 1, 2, 0 \rangle$ has one ascent and one descent.

Definition 22: Eulerian numbers

The symbol

$$\left\langle {n \atop k} \right\rangle \quad \textit{is the count of permutations of } n \textit{ elements with } k \textit{ ascents.}$$

$\left\langle {n \atop k} \right\rangle$ *is an Eulerian number and it is pronounced "n ascent k."*

From the above example, you know

$$\left\langle {3 \atop 0} \right\rangle = 1, \qquad \left\langle {3 \atop 1} \right\rangle = 4, \qquad \left\langle {3 \atop 2} \right\rangle = 1$$

10.1 How many permutations of n values have no ascents?

10.2 How many permutations of n values have n ascents?

10.3 How many permutations of n values have $(n-1)$ ascents?

10.4 Explain why

$$\sum_{k=0}^{n-1} \left\langle {n \atop k} \right\rangle = n!$$

10.5 Explain why

$$\left\langle {n \atop k} \right\rangle = \left\langle {n \atop n-k-1} \right\rangle$$

10.6 Fill in Euler's triangle listing the values of $\left\langle {n \atop k} \right\rangle$ for $n = 0, 1, 2, 3, 4, 5, 6$ and $k = 0, \ldots, n$.

The order of terms in the sum is not relevant.

11 A partition of positive integer n is a set of positive integers whose sum is n. For instance, $6 = (1+5) = (2+4) = (3+3)$ are three different partitions of 6 into 2 terms.

Let $p(n, k)$ denote the number of partitions of n into exactly k terms. For example, 6 can be partitioned into $k = 1, 2, 3, 4, 5$ and 6 terms in the following ways.

$p(6, 1) = 1$	(6)	1 way
$p(6, 2) = 3$	$(5+1, 4+2, 3+3)$	3 ways
$p(6, 3) = 3$	$(4+1+1, 3+2+1, 2+2+2)$	3 ways
$p(6, 4) = 2$	$(3+1+1+1, 2+2+1+1)$	2 ways
$p(6, 5) = 1$	$(2+1+1+1+1)$	1 way
$p(6, 6) = 1$	$(1+1+1+1+1+1)$	1 way

11.1 What are the boundary conditions $p(n, 1)$, $n = 1, 2, 3, \ldots$?

11.2 What are the boundary conditions $p(n, n)$, $n = 1, 2, 3, \ldots$?

11.3 Explain why the recurrence equation

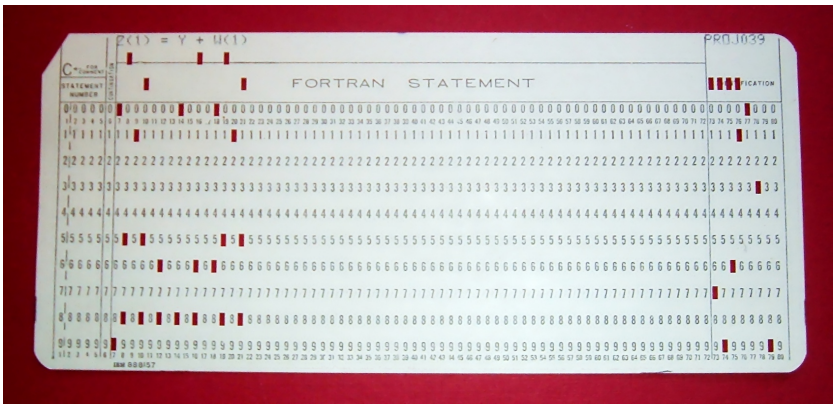
$$p(n, k) = p(n-1, k-1) + p(n-k, k)$$

describes how you could compute $p(n, k)$? Hints: How can you write n using k terms using a representation of $(n-1)$ with $k-1$ terms? How

can you write n using k terms using a representation of $(n-k)$ with k terms? Why must these two methods produce different sums?

11.4 Fill in "Partition into Terms" triangle listing the values of $p(n, k)$ for $n = 0, 1, 2, 3, 4, 5, 6$ and $k = 0, \ldots, n$.

12 Hollerith (punched) cards were used to write control command or computational data for a computer until the mid-1980's.



The IBM card, designed in 1928 has 12 rows and 80 columns. Assume there can be only one punch is allowed per column.

12.1 In how many ways can a card not be punched?

12.2 In how many ways can a card be punched once?

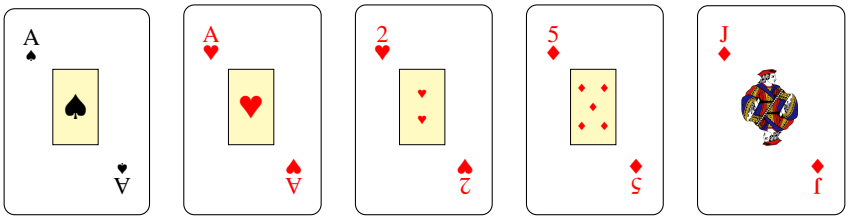
12.3 In how many ways can a card be punched twice?

12.4 In how many ways can a card be punched eighty times?

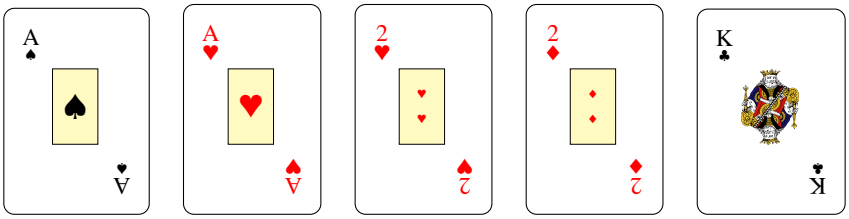
12.5 In how many ways can a card be (legitimately) punched eighty-one times?

13 A hand of five cards is dealt from a standard deck of 52 cards?

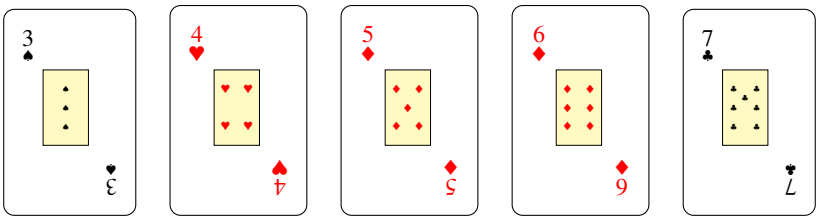
13.1 How many one-pair hands are there? An example one-pair hand is



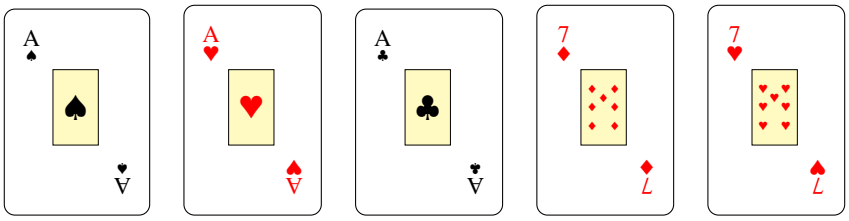
13.2 How many two-pair hands are there? An example two-pair hand is



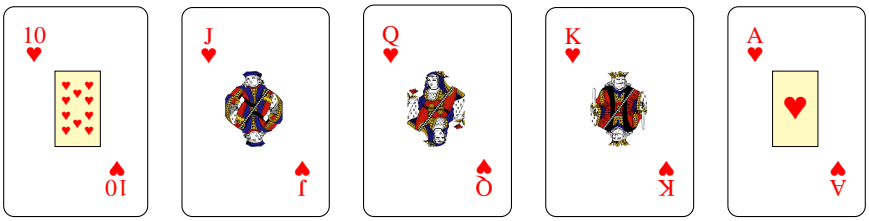
13.3 How many straight hands are there? An example straight hand is



13.4 How many full-house hands are there? An example full-house hand is



13.5 How many royal flush hands are there? An example royal flush hand is



13.6 What is the probability of each of the above classes of hands?

11. *Mathematical induction: Or is it recursion?*

Finally, you have broader considerations that might follow what you would call the "falling domino" principle. You have a row of dominoes set up, you knock over the first one, and what will happen to the last one is the certainty that it will go over very quickly. So you could have a beginning of a disintegration that would have the most profound influences.

Dwight D. Eisenhower, April 7, 1954 news conference

In(tro)duction

Mathematical induction works "bottom up." Starting from base cases (initial conditions), use previous results to generate the next result. The fundamental example is the construction of the natural numbers. Starting at 0, increment each number by 1 at each step generating the Gauss sequence $\vec{G} = \langle 0, 1, 2, 3, 4, \ldots \rangle$.

Interestingly, you can generate the natural numbers by establishing a natural transformation between them and sets. Specifically, let

- $0 \equiv \emptyset$, the set with cardinality 0.
- $1 \equiv 0 \cup \{0\} \equiv \{\emptyset\}$, a set with cardinality 1.
- $2 \equiv 1 \cup \{1\} \equiv \{\emptyset, \{\emptyset\}\}$, a set with cardinality 2.
- $3 \equiv 2 \cup \{2\} \equiv \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$, a set with cardinality 3.
- In general, the number n is associated with a set with n elements.

$$n \equiv (n-1) \cup \{n-1\}$$

Another example: Let $\tau(n)$ be the predicate statement "the sum of the first n natural numbers is $n(n-1)/2$."

$$\tau(n) :: 0 + 1 + 2 + \cdots + (n-1) = \frac{n(n-1)}{2} \qquad \tau \text{ is for "triangular."}$$

A mathematical induction proof shows that $\tau(n)$ is `True` for all natural numbers n.

$$(\forall n \in \mathbb{N})(\tau(n) = \texttt{True})$$

On the other hand, *recursion* works "top down." Reduce the computation of the result you need to the computation of smaller cases, and keep doing so until you reach terminal conditions (base cases).

Read the double colon "::" as "has the type". In this case $\tau(n)$ has type Boolean since the equality is either `True` or `False`.

Note that for $n = 0$ the sum

$$0 + 1 + \cdots (0-1) = 0 + 1 + \cdots + (-1)$$

is *non-sensical*. I say it is the empty sum, and equal to 0. That's good because the formula $n(n-1)/2$ is also equal to 0 at $n = 0$: Establishing a basis: The sum is equal to the formula at $n = 0$.

More explicitly,

$$\boxed{(\forall n \in \mathbb{N})\left(0+1+2+\cdots+(n-1)=\frac{n(n-1)}{2}\right)}$$

Mathematical induction has two steps, which I like to break into three parts. To prove that a predicate $\tau(n)$ is `True` for all natural numbers n do this:

1. Establish a *basis* for induction: Prove that $\tau(k)$ is `True` for some natural number k. Usually, $k=0$ or $k=1$ or some other small number.

$$\boxed{\text{Show } \tau(k) \text{ is } \texttt{True} \text{ for some natural number } k}$$

2. Complete the *inductive (two) step*:
   (a) Make an *inductive hypothesis*: Assume $\tau(n)$ is `True` for some natural number $n \geq k$.

$$\boxed{(\exists\, n,\ n \geq k)}\left(0+1+2+\cdots+(n-1)=\frac{n(n-1)}{2}\right)$$

   (b) Derive the *inductive conclusion*: Prove $\tau(n+1)$ is `True`. That is, prove

$$\boxed{(0+1+2+\cdots+(n-1))+n=\frac{(n+1)n}{2}}$$

Ah! The hypothesis is `True` for $n=k$. That was proven in the *basis*.

The *inductive step* establishes that the implication

$$\tau(n) \Rightarrow \tau(n+1) \quad \text{is } \texttt{True}.$$

If both steps can be demonstrated, then you can conclude that $\tau(n)$ is `True` for all natural numbers $n \geq k$.

The following two examples are fundamental: (1) Sums of natural numbers are binomial coefficients; (2) Sums of powers of 2 are Mersenne numbers. Learning how to employ mathematical induction to prove these two facts will help you understand the structure of other mathematical induction proofs.

1. The sum of the first n natural numbers can be evaluated by the formula $n(n-1)/2$, which is a Triangular number and the binomial coefficient $\binom{n}{2}$.

$$\boxed{0+1+2+\cdots+(n-1)=\frac{n(n-1)}{2}=\binom{n}{2}} \tag{13}$$

2. The sum of the first n powers of 2 can be evaluated by the formula 2^n-1, which is a Mersenne number.

$$\boxed{2^0+2^1+2^2+2^3+\cdots+2^{n-1}=2^n-1} \tag{14}$$

Let's see how inductive proofs of statements 13 and 14 go.

The sum $0+1+2+\cdots+(n-1)$ can be written in summation notation.

$$\sum_{k=0}^{n-1} k = 0+1+2+\cdots+(n-1)$$

When establishing a sum formula, I enjoy starting basis with no terms: $n=0$. In this case, the sum is empty with no terms and value 0. And, the function that computes the sum is $0 \to \frac{0(0-1)}{2}$ also 0.

You know the implication is `True` when $\tau(n)=$ `False`. There is nothing to prove in this case.

$(n+1)n/2$ is the expression of the function $n \to n(n-1)/2$ when n is replaced by $(n+1)$.

Theorem 24: Sum of natural numbers

The sum of the first n natural numbers

$$0 + 1 + 2 + 3 + \cdots + (n-1)$$

can be evaluated by computing the value of the Triangular function

$$n \to \frac{n(n-1)}{2} = \binom{n}{2}$$

Proof: Sum of natural numbers

The proof has these steps:

1. *The* basis *for induction can start at* $n = 1$. *In this case, there is only one term in the sum: The sum is simply* 0. *And, the function is* $1 \to \frac{1(1-1)}{2} = 0$. *The sum is equal to the function value at* $n = 1$.

2. *The* inductive step *shows the implication*

$$\tau(n) \Rightarrow \tau(n+1) \quad \textit{is } \texttt{True}.$$

   (a) Assume $\tau(n) = \texttt{True}$ *for some value of* $n \geq 1$. *That is, assume*

$$(\exists n \in \mathbb{N},\ n \geq 1)\left(0 + 1 + 2 + \cdots + (n-1) = \frac{n(n-1)}{2}\right)$$

   (b) Prove *the sum of the first* $(n+1)$ *natural numbers equals* $(n+1)n/2$.
   To do so, note that the sum of the first $(n+1)$ *natural numbers is the sum of the first* n *plus the* $(n+1)^{st}$ *natural number. This leads to the derivation:*

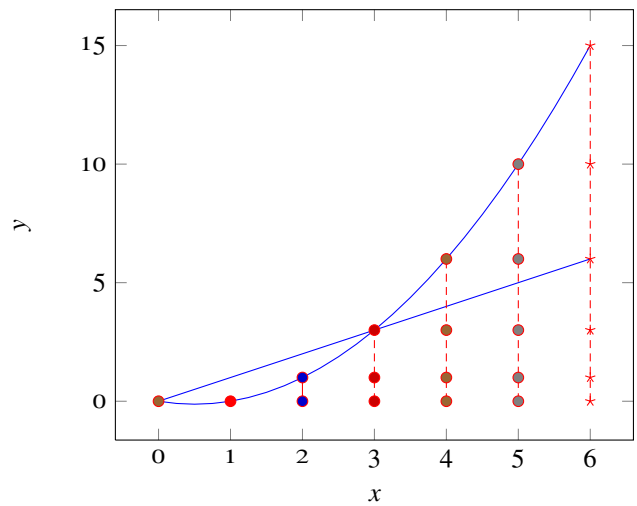
$$\begin{aligned}
[0 + 1 + 2 + 3 + \cdots + (n-1)] + n &= \frac{n(n-1)}{2} + n \\
&= \frac{n(n-1)}{2} + \frac{2n}{2} \\
&= \frac{(n^2 - n) + 2n}{2} \\
&= \frac{(n^2 + n)}{2} \\
&= \frac{(n+1)n}{2} \\
&= \tau(n+1)
\end{aligned}$$

Therefore, $\tau(n+1) = \texttt{True}$ *for all values of* n. *That is,*

$$(\forall n \in \mathbb{N})\left(0 + 1 + 2 + \cdots + (n-1) = \frac{n(n-1)}{2}\right)$$

The graphs below of $y(x) = x(x-1)/2$ and the sum of values on the graph of $\triangle y(x) = y(x+1) - y(x) = x$ show the values agree.

This may be confusing. The red dots along a vertical dashed line represent 0, 1, 2, 3, 4 and so on. The space between the dots are (dashed) line segments of these lengths stacked upon one another. The sum of lengths alway reaches the blue quadratic curve.



Now, let's tackle statement 13 from above.

The sum $2^0 + 2^1 + 2^2 + \cdots + 2^{n-1}$ can be written in summation notation.

$$\sum_{k=0}^{n-1} 2^k = 2^0 + 2^1 + 2^2 + \cdots + 2^{n-1}$$

Theorem 25: Sum of powers of 2

The sum of the first n powers of 2

$$2^0 + 2^1 + 2^2 + 2^3 + \cdots + 2^{n-1}$$

can be evaluated by computing the value of the function

$$n \to 2^n - 1$$

I enjoy starting the basis at $n = 0$. In this case, interpret the sum to be empty with value 0. And, the function $0 \to 2^0 - 1 = 0$ also.

2^{n+1} is the expression of the function $n \to 2^n - 1$ when n is replaced by $(n+1)$.

Proof: Sum of powers of 2

The proof has these steps:

1. *The* basis *for* induction *can start at* $n = 1$. *In this case, the sum is simply* $2^0 = 1$, *and the* function *is* $1 \to 2^1 - 1 = 1$ *also. The sum is equal to the value of the* function *at* $n = 1$.

2. *The* inductive step *shows the implication*

$$\tau(n) \Rightarrow \tau(n+1) \quad \textit{is } \texttt{True}.$$

   (a) Assume *the value of the sum (of the first* n *powers of* 2*) is equal to the value of the* function $2^n - 1$, *for some* $n \geq 1$. *That is,*

$$(\exists n \in \mathbb{N},\ n \geq 1)(2^0 + 2^1 + 2^2 + \cdots + 2^{n-1} = 2^n - 1)$$

   (b) Prove *the sum (of the first* $(n+1)$ *powers of* 2*) is the same as the value* $2^{n+1} - 1$.

To do so, note the sum of the first $(n+1)$ powers of 2 is the sum of the first n powers of 2 plus the $(n+1)^{st}$ power of 2. This leads to the derivation:

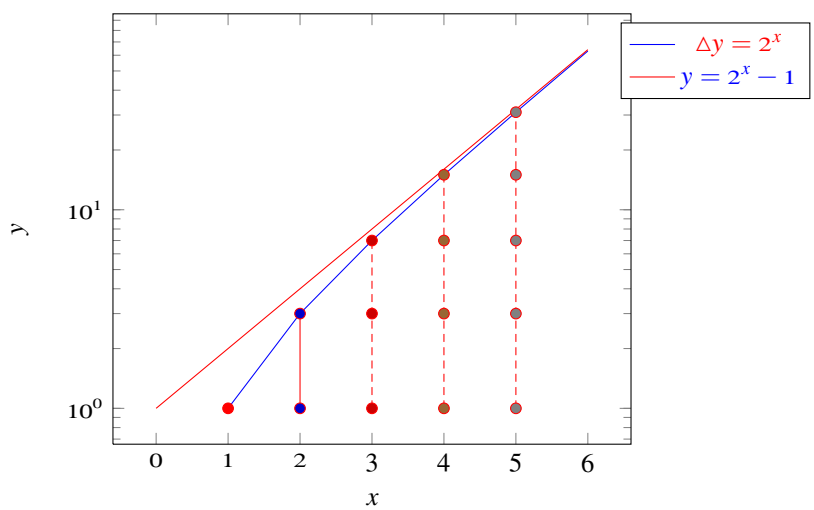
$$\begin{aligned}(2^0+2^1+2^2+2^3+\cdots+2^{n-1})+2^n &= (2^n-1)+2^n\\ &= 2\cdot 2^n-1\\ &= 2^{n+1}-1\end{aligned}$$

Therefore, the statement is `True` *for all values of n.*

$$(\forall n\in\mathbb{N})\left(1+2+4\cdots+2^{n-1}=2^n-1\right)$$

Notice the graph of $y(x)=2^x-1$ and the sum of values on the graph of $\triangle y(x)=y(x+1)-y(x)=2^x$ agree. The axes in the graph are semi-log. That is, the x axis is linear, while the y axis is plotted logarithmically.

Wow! This is even more confusing, because the y-axis is compressed by a logarithmic scaling. Instead of plotting x versus y, The graph plots x versus $\log y$. Because 2^x is just a little bigger than 2^x-1 the two curves appear to overlap as x becomes large, but they are always 1 apart.



Notice that adding up red points gives a value on the graph.

There are many other useful statements that have simple inductive proofs. Too many count. A few will be mentioned.

Geometric sums

You can generalize the statement about sums of powers of 2 equaling a function to arbitrary geometric sequences. That is, you can find the function that evaluates an sums of terms in a geometric sequence.

For instance, consider sums of powers of 3: Sums the terms in the sequence

$$\left\langle 1,\ 3,\ 9,\ 27,\ 81,\ 243,\ldots,\ 3^{n-1},\ldots\right\rangle$$

The first few sums are:

$$\begin{aligned}
\sum_{0}^{-1} = 0 &= \frac{3^0 - 1}{2} && \text{(no terms (empty sum))} \\
1 = 1 &= \frac{3^1 - 1}{2} && \text{(one term)} \\
1 + 3 = 4 &= \frac{3^2 - 1}{2} && \text{(two terms)} \\
1 + 3 + 9 = 13 &= \frac{3^3 - 1}{2} && \text{(three terms)} \\
&\vdots \\
1 + 3 + \cdots + 3^{n-1} &= \frac{(3^n - 1)}{2} && (n \text{ terms})
\end{aligned}$$

Do you see a pattern? Here's the general statement.

Theorem 26: Geometric sums

The sum of terms in a geometric sequence is $a(r^n - 1)/(r - 1)$. That is, let

$$\vec{R} = \langle a, ar, ar^2, ar^3, \ldots \rangle = \langle r_0, r_1, r_2, r_3, \ldots \rangle$$

be a geometric sequence, with *ratio* $r \neq 0$ and *initial condition* $r_0 = a$. Then, the sum of the first n terms in $\vec{R}$ is

$$a + ar + ar^2 + \cdots + ar^{n-1} = \sum_{k=0}^{n-1} ar^k = a\left(\frac{r^n - 1}{r - 1}\right) \tag{15}$$

Proof: Geometric sums

The proof that equation 15 is `True` *for all natural numbers goes like this.*
For $n = 1$ both sides of equation 15 equal a. Next, if the equation is `True` *for some $n \geq 1$, then you can derive*

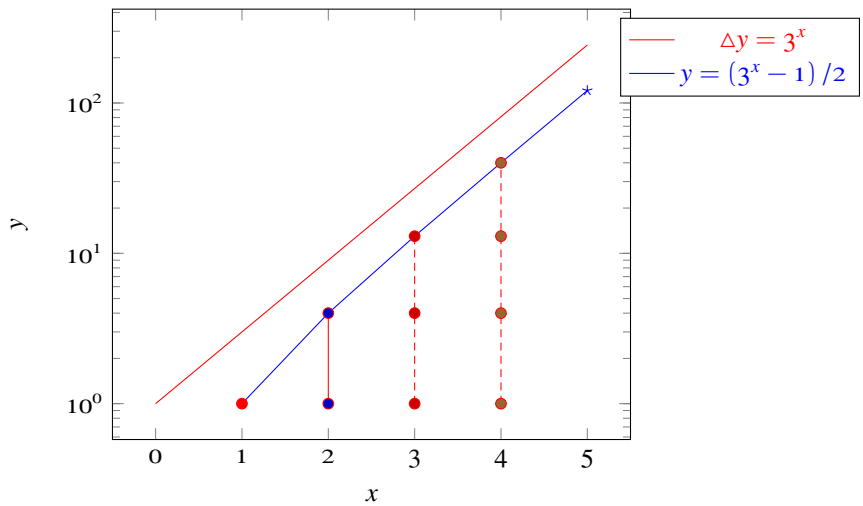
$$\begin{aligned}
(a + ar + ar^2 + \cdots + ar^{n-1}) + ar^n &= a\left(\frac{r^n - 1}{r - 1}\right) + ar^n \\
&= a\left(\frac{r^n - 1}{r - 1} + \frac{r^n(r - 1)}{(r - 1)}\right) \\
&= a\left(\frac{r^n - 1 + r^{n+1} - r^n}{r - 1}\right) \\
&= a\left(\frac{r^{n+1} - 1}{r - 1}\right)
\end{aligned}$$

Therefore, if the statement is `True` *for some n, then it is* `True` *for $(n + 1)$ as well. Using this you can conclude the statement of the theorem is* `True` *for all values of n.*

The graphs below are of $y(x) = (3^x - 1)/2$ and the sum of values on the

graph of $\Delta y(x) = y(x+1) - y(x) = 3^x$. Notice that adding up red points gives a value on the graph.

The axes in the graph are *semi-log*. That is, the x axis is linear, while the y axis is plotted logarithmically.



In HASKELL, the code to sum the terms in an geometric sequence might look like this.

```
geometricSum :: [Arithmetic] -> Integer
geometricSum [] = 0
geometricSum  (ns) = head ns * (head ns + last ns)/2
```

```
SumNats :: [Geometric] -> Integer
SumNats [] = 0
SumNats (ns)
   = let r = (head head ns)/head ns
   in head ns * (r ** length ns -1)/(r-1)
```

Arithmetic sums

You can generalize the statement about sums of natural numbers equaling a particular function to an arbitrary arithmetic sequences. That is, you can find the function that evaluates an sums of terms in an arithmetic sum. Consider an arithmetic sequence that starts at -2 and increased by 3 at each step

$$\langle -2, 1, 4, 7, 10, 13, \ldots \rangle$$

$$\sum_{0}^{-1} = 0 = 0 \cdot \frac{\cdot + \cdot}{2} \quad \text{(no terms (empty sum))}$$
$$-2 = -2 = 1 \cdot \frac{-2-2}{2} \quad \text{(one term)}$$
$$-2+1 = -1 = 2 \cdot \frac{-2+1}{2} \quad \text{(two terms)}$$
$$-2+1+4 = 3 = 3 \cdot \frac{-2+4}{2} \quad \text{(three terms)}$$
$$-2+1+4+7 = 10 = 4 \cdot \frac{-2+7}{2} \quad \text{(four terms)}$$
$$\vdots$$
$$-2+1+\cdots+(3n-5) = n \cdot \frac{-2+(3n-5)}{2} = n \cdot \frac{3n-7}{2} \quad (n \text{ terms})$$

Do you see a pattern? The sum is the number of terms n times the average of the first and last term in the sum.

Here's the general statement.

Theorem 27: Arithmetic Sums

Let $\vec{A} = \langle a_0, a_1, a_2, a_3, \ldots \rangle = \langle b, m+b, 2m+b, 3m+b, \ldots \rangle$ be an arithmetic sequence.
The sum of the first n terms of $\vec{A}$ is

$$\sum_{k=0}^{n-1} (km+b) = n\left(\frac{2b+(n-1)m}{2}\right) = (\#\text{ terms})\frac{1^{\text{st}} + n^{\text{th}}}{2}$$

Proof: Arithmetic sums

Here's how a proof might be stated.
The basis for induction is: The sum of no ($n = 0$) terms is empty and equal to 0. Likewise, the function value

$$n \to n\left(\frac{2b+(n-1)m}{2}\right) \quad \textit{is } 0 \textit{ when } n = 0$$

Now, assume

$$\sum_{k=0}^{n-1} (km+b) = n\left(\frac{2b+(n-1)m}{2}\right) \quad \textit{is } \texttt{True} \textit{ for some } n \geq 0.$$

Then, *the sum of the first $(n+1)$ terms equals the sum of the first n terms plus the last ($(n+1)^{st}$) term. That is,*

$$\sum_{k=0}^{n} (km+b) = \left(\sum_{k=0}^{n-1} (km+b)\right) + (nm+b)$$

To see this, use the hypothesis to replace

$$\sum_{k=0}^{n-1}(km+b) \quad \text{with} \quad n\left(\frac{2b+(n-1)m}{2}\right)$$

and follow the algebra.

$$\begin{aligned}
\sum_{k=0}^{n}(km+b) &= \left[\sum_{k=0}^{n-1}(km+b)\right] + (nm+b) \\
&= n\left(\frac{2b+(n-1)m}{2}\right) + (nm+b) \\
&= \left(\frac{2nb+n(n-1)m}{2}\right) + \frac{2nm+2b}{2} \\
&= \frac{2nb+n(n-1)m+2nm+2b}{2} \\
&= \frac{2b(n+1)+n(n+1)m}{2} \\
&= (n+1)\left(\frac{2b+nm}{2}\right)
\end{aligned}$$

Therefore, if the statement is `True` *for some* n*, then it is* `True` *for* $(n+1)$ *as well. Using this you can conclude the statement of the theorem is* `True` *for all values of* n*.*

In HASKELL, the code to sum the terms in an arithmetic sequence might look like this.

```
arithmeticSum :: [Arithmetic] -> Integer
arithmeticSum [] = 0
arithmeticSum (ns) = length ns * (head ns + last ns)/2
```

Pass a Quiz: Basic induction problems

Take a quiz on page 327 to check your understanding. You can return to here from the quiz.

Strong induction

Sometimes the inductive hypothesis needs to be strengthened. For instance, you may need to know that all of

$$\text{P}(k),\ \text{P}(k+1),\ \text{P}(k+2), \ldots,\ \text{P}(n-1) \quad \text{are } \texttt{True}$$

in order to conclude $\text{P}(n)$ is true. A classical example is the fundamental theorem of arithmetic. The theorem was know to Euclid and is also called the *Unique factorization theorem*.

Theorem 28: Fundamental theorem of arithmetic

Every integer n greater than 1 is either a prime number or the product of prime numbers. The factorization is unique, except for the order of the factors.

Proof: Fundamental theorem of arithmetic

As a basis for induction: Each of $n = 2, 3, 4 = 2^2, 5, 6 = 2 \cdot 3$ can be written uniquely as a product of prime numbers. Make the hypothesis that, for some $n \geq 7$, all natural numbers from 2 to $n - 1$ can be written uniquely as the product of primes. Consider n, the next value after $n - 1$, If n is a prime number, the proof is complete. If n is not prime, then it can be written as the product of two integers $n = ab$ where $1 < a \leq b < n$. By the inductive hypothesis, both a and b can be written as the product of primes. Therefore, n can be written as the product of primes. Establishing the uniqueness of the factorization is best accomplished using a contradiction.

Perhaps Dorothy, from the Wizard of Oz, asking the Guardian of the Emerald City Gates: "What kind of a horse is that?" and receiving the reply "He's the Horse of a Different Color, you've heard tell about." led George Polya to his statement that "there is no horse of a different color". His `False` *proof* shows what can go wrong when you need strong induction and don't use it.

Theorem 29: False theorem: No horse of a different color

All horses are of the same color.

Proof: No horse of a different color

You are to find the mistake in this "proof."
If there is only one horse, it is of the same color as itself. Assume, for some n, that any set of n horses all have the same color. Let $\mathbb{X}$ be a set of $n + 1$ horses. By the hypothesis, the first n and last n horses all have the same color. And, since the two sets overlap, all horses in $\mathbb{X}$ have the same color.

Homework Questions

Use your time outside of class to solve these problems.

1 Consider the first few sums of powers on k for $k = 0, 1, 2, 3, \ldots, (n-1)$. Use mathematical induction to show that:

1.1 $0 + 1 + 2 + 3 + \cdots + (n-1) = \sum_{k=0}^{n-1} k = \frac{n(n-1)}{2}$

1.2 $0^2 + 1^2 + 2^2 + 3^2 + \cdots + (n-1)^2 = \sum_{k=0}^{n-1} k^2 = \frac{n(n-1)(2n-1)}{6}$

1.3 $0^3 + 1^3 + 2^3 + 3^3 + \cdots + (n-1)^3 = \sum_{k=0}^{n-1} k^3 = \frac{n^2(n-1)^2}{4}$

2 Consider the following inductive proof of the statement: "Every odd natural number is prime." Proof: 3, 5 and 7 are prime numbers. Therefore the statement is `True`. Do you find anything wrong with this proof?

3 The notes make that statement "the sum of natural numbers is a binomial coefficient." Establish that the general equation below is always `True`

$$\sum_{k=m}^{n-1} k = \binom{n}{2} - \binom{m}{2} \quad \text{for } n \geq m.$$

4 The notes make that statement "the sum of powers of 2 is a Mersenne number." Establish that the general equation below is always `True`

$$\sum_{k=m}^{n-1} 2^k = 2^m(2^{n-m} - 1)$$

5 If you invest A dollars each month and you earn interest at a monthly rate on the total amount invested, what will your investment be worth after n months?

6 Use mathematical induction to prove the sum of the even natural numbers from 0 to $2n$ is $n(n+1)$.

$$\sum_{k=0}^{n} 2k = n(n+1)$$

7 Use mathematical induction to prove the sum of products consecutive pairs of natural numbers is the product of three consecutive number divided by 3, that is,

$$\sum_{k=0}^{n-1} k(k-1) = \frac{n(n-1)(n-2)}{3}$$

8 Use mathematical induction to prove the summation formula

$$\sum_{k=1}^{n} \frac{1}{(2k-1)(2k+1)} = \frac{n}{2n+1}$$

is true for all natural numbers $n \geq 0$.

9 Use mathematical induction to prove the sum of k times $k!$ from $k = 0$ to $k = n-1$ is $n! - 1$

$$\sum_{k=0}^{n-1} k \cdot k! = n! - 1.$$

An interesting identity involving π is

$$\frac{\pi^2}{6} = \sum_{k=1}^{\infty} \frac{1}{k^2}$$

10 Use mathematical induction to prove the summation formula

$$\sum_{k=1}^{n} \frac{1}{k^2} < 2 - \frac{1}{n}$$

is true for any natural number $n \geq 2$.

11 Prove that the sum of binomial coefficients in a column of Pascal's triangle equals the binomial coefficient in the next column and down one row from where you stopped the sum. That is, show that

$$\sum_{k=0}^{n-1} \binom{k}{m} = \binom{n}{m+1}$$

12 Prove that the sum of all binomial coefficients in a row of Pascal's triangle equals a power of 2, that is,

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n$$

How is this related to the total number of subsets of an n-element set? How is this related to the cardinality of the power set of the set?

13 Use mathematical induction to show that the sum of the first n Fibonacci numbers is a Fibonacci number minus one. Specifically, show that

$$F_0 + F_1 + \cdots + F_{n-1} = F_{n+1} - 1$$

14 Prove that the sum of the odd-indexed Fibonacci numbers up to F_{2n+1} equals F_{2n+2}. That is,

$$\sum_{k=0}^{n} F_{2k+1} = F_{2n+2}$$

15 Prove that

$$\sum_{k=0}^{n} F_{2k} = F_{2n+1} - 1$$

16 Prove that

$$\binom{n}{0} + \binom{n-1}{1} + \binom{n-2}{2} + \cdots + \binom{0}{n} = F_{n+1}$$

17 Prove 6 divides $7^n - 1$ for all natural numbers n.

18 Prove $n! \geq 2^n$ for all natural numbers $n \geq 4$.

19 An Indian folk tale tells Rani, a peasant girl who did a good deed. As reward, the Raja said he would grant any wish. Rani only asked for one grain of rice today, two grains tomorrow, four grains the next day, and so on for 30 days. The Raja granted Rani's wish. Assuming none of the rewarded rice was consumed

19.1 How much rice did Rani have after 30 days?

19.2 The Freerice website estimates 20,000 grains of rice are needed (along with other ingredients) to feed a person for one day. How many people could Rani feed for only 1 day? For 2 days?, For 3 days?

19.3 What function maps the store in Rani's rice warehouse to the number of people who could be feed for n days?

20 What is wrong in the proof that there is no horse of a different color?

21 Prove Bernoulli's inequality: If $1 + h > 0$ then $1 + nh \leq (1 + h)^n$.

22 Consider the table

$$\begin{aligned} 1 &= 0 + 1 \\ 2 + 3 + 4 &= 1 + 8 \\ 5 + 6 + 7 + 8 + 9 &= 8 + 27 \\ 10 + 11 + 12 + 13 + 14 + 15 + 16 &= 27 + 64 \end{aligned}$$

Guess the general rule these instances suggest and prove your rule is correct by using mathematical induction.

23 The Zeno is named for Zeno of Elea, the Greek philosopher, who argued that an arrow could never reach its mark: It must reach its halfway point before the mark and there is always a next halfway point. What is the sum of first n halfway distances?

24 I started an annuity 40 years ago. Each month i have deposited \$100 into an account with a rate of return of one-half of one-percent per month on the value of the account.

1. How much money have I paid into the account?

2. With a little bit of analysis, you can deduce that the value of my account can be computed by the sum

$$100\left[1 + 1.005 + (1.005)^2 + \cdots + (1.005)^{479}\right] \quad \text{dollars}$$

Use you knowledge of geometric sums to my account value is approximately $\$20000(1.005)^{480}$, about \$200,000.

25 Curious George wondered if the sum formulas

$$\sum_{k=0}^{n-1} 1 = \binom{n}{1} \text{ and } \sum_{k=0}^{n-1} k = \binom{n}{2} \text{ generalize to } \sum_{k=0}^{n-1} k^2 = \binom{n}{3}$$

Show that CG's conjecture is wrong by giving a simple counterexample.

26 The correct formula for the sum of squares is

$$\sum_{k=0}^{n-1} k^2 = \binom{n}{3} + \binom{n+1}{3}$$

Use mathematical induction to prove this equation is valid.

27 It is interesting to write Pascal's triangle as a triangle the way most people do.

1												
1	1											
1	2	1										
1	3	3	1									
1	4	6	4	1								
1	5	10	10	5	1							
1	6	15	20	15	6	1						
1	7	21	35	35	21	7	1					
1	8	28	56	70	56	28	8	1				
1	9	36	84	126	126	84	36	9	1			
1	10	45	120	210	252	210	120	45	10	1		
1	11	55	165	330	462	462	330	165	55	11	1	
1	12	66	220	495	792	924	792	495	220	66	12	1

27.1 Consider the sequence of the central column

$$\langle 1, 2, 6, 20, 70, 252, 924, \ldots \rangle = \langle c_0, c_1, c_2, c_3, c_4, c_5, c_6, \ldots \rangle$$

What function (as a binomial coefficient) computes terms c_n in this sequence?

27.2 What recurrence equation does the function satisfy?

27.3 Show that c_n can be computed by the sum

$$\sum_{k=0}^{n} \binom{n}{k}^2$$

12. Recursion: Or is it mathematical induction?

Hofstadter's Law: It always takes longer than you expect, even when you take into account Hofstadter's Law.

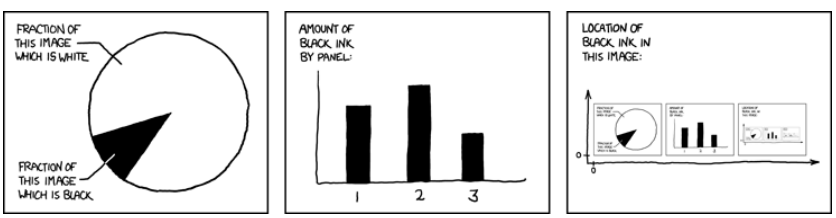
Douglas Hofstadter, Gödel, Escher, Bach: An Eternal Golden Braid (Hofstadter, 1999)

[Recursive] Recursive: adj. See: Recursive

Stan Kelly-Bootle, English computer scientist and author (1929 –). Devil's DP Dictionary. (Kelly-Bootle, 1981)

Compound interest is the eighth wonder of the world. He who understands it, earns it . . . he who doesn't . . . pays it.

Albert Einstein



Self-Description

Examples of recursion

Recursion works "top down:" Reduce a complex problem to the computation of smaller cases of the same problem. Keep doing so, recursively, until you reach terminating conditions [base cases]. Recursion defines objects in terms of *terminal conditions* and *rules*.

On the other hand, induction works "bottom up." Starting from base cases (initial conditions), use previous results to generate the next result.

1. One or more *terminal conditions* [base case(s)]
2. *Rules* that reduce all other cases to the terminal conditions [base case(s)].

Here are some examples of recursive definitions.

Addition on the natural numbers

Let n be a natural number. *Assume* the existence of $\text{succ}(n)$, the successor function, that computes $\text{succ}(n) = (n+1)$ for any natural numbers n. Then, $\text{add}(n, a) = (n+a)$ can be defined by:

The *successor* function is postulated to exist as one of the Peano axioms for arithmetic on the natural numbers.

- The *base case* is $\text{add}(0, a) = a$.
- The *rule* is $\text{add}(\text{succ}(n), a) = \text{succ}(\text{add}(n, a))$ For instance,

$$\begin{aligned}
\text{add}(0, a) &= a \\
\text{add}(1, a) &= \text{succ}(\text{add}(0, a)) = \text{succ}(a) = a+1 \\
\text{add}(2, a) &= \text{succ}(\text{add}(1, a)) = \text{succ}(a+1) = a+2 \\
\text{add}(3, a) &= \text{succ}(\text{add}(2, a)) = \text{succ}(a+2) = a+3 \\
&\vdots \quad \vdots
\end{aligned}$$

Summations

1. The *base case* for summations: "The sum of no terms, the empty sum $\sum_0^{-1}$, has value 0."

$$\sum_{k=0}^{-1} \cdot = 0 \quad \text{where } \cdot \text{ is any expression.}$$

   Recall the notes on summations and the use of notation to denote the sum of no terms.

2. The *rule* is "the sum of the first $(n+1)$ terms is the sum of the first n terms plus the missing $(n+1)^{\text{st}}$ term."

$$\sum_{k=0}^{n} f(k) = \left[\sum_{k=0}^{n-1} f(k)\right] + f(n)$$

You could rewrite the recursive rule for sums as: "the sum of n terms is the first term plus the sum of last $(n-1)$ terms."

$$\begin{aligned}
f(0) + f(1) + \cdots + f(n-1) &= \sum_{k=0}^{n-1} f(k) \\
&= f(0) + \left[\sum_{k=1}^{n-1} f(k)\right]
\end{aligned}$$

Writing the summation notation is tiring. You could define $S(0:n)$ to be the sum from $k=0$ to $k=(n-1)$, then the recurrence equation is

$$S(0:n) = f(0) + S(1:n)$$

You could also write S_n for the sum of the first n terms. Then the recursion is

$$S_n = S_{n-1} + S_{n-1}$$

This rule for sums was used extensively in the notes about induction. For instance, in proving $t(n) = n(n-1)/2$ computes the (partial) sums of natural numbers.

$$0 + 1 + 2 + \cdots + (n-1) + n = [0 + 1 + 2 + \cdots + (n-1)] + n$$

And, proving $m(n) = 2^n - 1$ computes the (partial) sums of powers of 2.

$$2^0 + 2^1 + \cdots + 2^{n-1} + 2^n = \left[2^0 + 2^1 + \cdots + 2^{n-1}\right] + 2^n$$

Factorials

A factorial is a product of factors. The *base case* is $0! = 1$. The *rule* is

$$\boxed{n! = n \cdot (n-1)! \quad \text{for } n > 0.}$$

Factorials count the number of permutations on a set of objects. People often ask: Why is $0! = 1$? Here are some reasons.

1. Nothing can be arranged (permuted) in 1 way: Just do nothing. There is only 1 way to do nothing: Don't do anything.

2. For the rule $n! = n(n-1)!$ to remain `True` for $n = 1$, it must be that

$$1 = 1! = 1(1-1)! = 0!$$

3. For the binomial coefficient $\binom{n}{k}$ to be sensical at $k = 0$ and $k = n$ it must be that $0! = 1$

$$\binom{n}{0} = \frac{n!}{0!n!} = \binom{n}{n} = 1$$

Stirling's approximations to $n!$ is interesting.

Theorem 30: Stirling's sapproximation to $n!$

n factorial is *asymptotically* approximated by

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

That is, the ratio approaches 1 as n increases without bound.

$$\lim_{n\to\infty} \frac{n!}{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n} = 1$$

You could use the more ornate notation

$$\prod_{k=1}^{n} k \quad \text{to denote } n!.$$

This "product" notation parallels "summation" notation. for example, the sum of the first n positive integers is

$$\sum_{k=1}^{n} k \quad \text{to denote } n(n+1)/2.$$

Can you define $(-1)!$? What about $(-2)!$?

Binomial coefficients

Recall the notes on computing the value of "n choose k" and counting subsets if you need to refresh your mind on the notion of binomial coefficients. For binomial coefficients the base cases are *boundary conditions*.

$$\boxed{\binom{n}{0} = \binom{n}{n} = 1 \quad \text{for } n \in \mathbb{N}.}$$

The *rule* is Pascal's identity,

Theorem 31: Pascal's identity

Binomial coefficients satisfy the recurrence

$$\binom{n}{m} = \binom{n-1}{m} + \binom{n-1}{m-1}, \quad \text{for all integers } m \text{ and } n > 0.$$

Unlike the previous examples, Pascal's identity is a 2-dimensional recursion: It involves two variables n and m.

Recursive definition of strings

Recursion is a fundamental computing concept. An understanding of recursion is useful for writing solutions to problems in a programming languages. Here are a few fundamental things you can define using recursion.

A *word* w is a string in a language $\mathbb{L} \subseteq \mathbb{A}^*$, pronounced "A-star," the Kleene closure of $\mathbb{A}$, the set of all strings that can be formed over the alphabet $\mathbb{A}$.

Strings over an alphabet $\mathbb{A}$ can be defined recursively.

Definition 23: Strings

A string s over $\mathbb{A}$ is defined by:

1. *The empty string λ is a string.*
2. *If s is a string and $c \in \mathbb{A}$ is a character, then cs is a string.*

Palindromes are a special type of string: They read the same forward and backward.

Some example palindromes are:

civic, radar, level, kayak

Able was I ere I saw Elba

Definition 24: Palindromes

A palindrome is a string defined by:

1. *The empty string λ is a palindrome.*
2. *Let $c \in \mathbb{A}$ be a character. Then c is a palindrome.*
3. *If $c \in \mathbb{A}$ is a character and s is a palindrome, then csc is a palindrome.*

Pass a Quiz: Basic problems on recursion

Take a quiz on page 327 to check your understanding. You can return to here from the quiz.

Recursive definitions of sequences

Many sequences can be recursively. Commonly, this is done by giving *initial* or *boundary conditions* and a recurrence equation. The notes on example sequences in the notes on the sum & difference calculus give several examples of sequences defined by an *initial condition* and a recurrence equation.

- The Alice sequence $\vec{A} = \langle 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, \ldots \rangle$ can be enumerated by the initial condition

$$a_0 = 1 \quad \text{initial condition}$$

and the recurrence equation

$$a_n = a_{n-1} \quad \text{for } n > 0.$$

The Alice sequence is the first column in Pascal's triangle. If you change the initial condition, but keep the equation different sequences are generated. For instance, if $a_0 = \pi$, then the "pie-sequence" is generated

$$\vec{\pi} = \langle \pi, \pi, \pi, \pi, \ldots \rangle$$

Summing terms in the Alice sequence gives the Gauss sequence.

- The Gauss sequence $\vec{G} = \langle 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 \ldots \rangle$ can be enumerated by

$$g_0 = 0 \quad \text{initial condition}$$

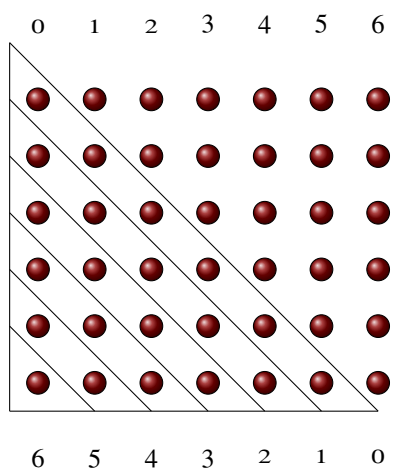
and the recurrence equation

$$g_n = g_{n-1} + 1 \quad \text{for } n > 0.$$

The Gauss sequence is the second column in Pascal's triangle.

If you draw lines around triangles, you'll see the sum of the natural numbers generates the triangular numbers: Stack crimson balls from the Gauss sequence as diagonals on smaller triangles produces another triangle.

Note the rectangle at the left is 7 dots by 6 dots. The lower triangle is one-half of the dots. The number of dots in the lower triangle is $(7 \cdot 6)/2 = 21$.



$$\sum_{0 \le k < 7} (6 - k) = 6 + 5 + 4 + 3 + 2 + 1 + 0 = 21 = \frac{7(7-1)}{2}$$

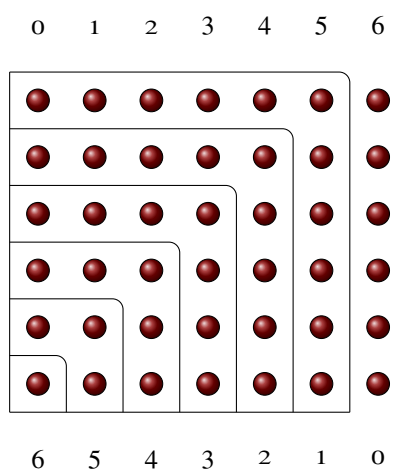
If you draw lines around squares, you'll see the sum of odd integers produces squared numbers n^2..

The study of single-variable recurrence equations and their solutions is the discrete analog of the study of differential equations from continuous mathematics. But we'll go more elementary.

You can handily show that $x = 3$ satisfies $x^2 + x - 12 = 0$. Simply substitute 3 for x in the equation and compute

$$3^2 + 3 - 12 = 9 + -12 = 0$$

It takes more skill to find 3 in the first place.



$$\sum_{0 \le k < 7} (2k+1) = 1+3+5+7+9+11+13 = 7^2$$

- The triangular sequence $\vec{T} = \langle 0, 0, 1, 3, 6, 10, 15, 21, 28, 36, 45, \ldots \rangle$ can be enumerated by

$$t_0 = 0 \quad \text{initial condition}$$

and the recurrence equation

$$t_n = t_{n-1} + (n-1) \quad \text{for } n > 0.$$

The triangular sequence is the third column in Pascal's triangle.

- Powers of 2 form the *basis* for writing numbers in binary notation.

$$\vec{P}_2 = \langle 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, \ldots \rangle$$

powers of 10 form the *basis* for writing numbers in decimal notations.

$$\vec{P}_{10} = \langle 1, 10, 100, 1000, 10000, 100000 \ldots \rangle$$

- The Mersenne sequence $\vec{M} = \langle 0, 1, 3, 7, 15, 31, 63, 127, 255, 511, \ldots \rangle$ can be enumerated by

$$m_0 = 0 \quad \text{initial condition}$$

and the recurrence equation

$$m_n = 2m_{n-1} + 1 \quad \text{for } n > 0.$$

Terms in the Mersenne sequence are sums of terms in the powers of 2 sequence.

$$m_n = \sum_{k=0}^{n-1} 2^k$$

- The Fibonacci sequence $\vec{F} = \langle 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, \ldots \rangle$ can be enumerated by

$$f_0 = 0 \quad \text{and} \quad f_1 = 1 \quad \text{initial conditions}$$

and the recurrence equation

$$f_n = f_{n-1} + f_{n-2} \quad \text{for } n > 1.$$

The Fibonacci recurrence is "second-order" meaning the previous 2 values are needed to compute the next term.

Changing the initial conditions to $f_0 = 2$ and $f_1 = 1$ generates the Lucas sequence

$$\vec{L} = \langle 2, 1, 3, 4, 7, 11, 18, 29, 47, 76, 123, \ldots \rangle$$

Functions satisfying recurrence equations

We want to show how a given function $f(n)$ satisfies given terminal condition(s) and some recurrence equation. Mathematical induction can be used to proof many useful results. Here are some examples.

The Triangular sequence

The function $t(n) = n(n-1)/2 = \binom{n}{2}$ satisfies the triangular recurrence

$$\boxed{t_n = t_{n-1} + (n-1) \quad \text{for } n > 0, \text{ with initial condition } t_0 = 0.} \tag{16}$$

The functional form $t(n)$ is just another name for the subscripted sequential form t_n.

Oh! We're going backwards now.

You 'could' start on the left with $t(n) = n(n-1)/2$ and massage it into the right side of the equation, but I think that would be harder.

The algorithmic idea behind the triangular recurrence is: To solve a problem of size n you must solve a similar problems of size $(n-1)$ and then perform $(n-1)$ more steps.

This recursion generates the triangular sequence

$$\boxed{\vec{T} = \langle 0, 0, 1, 3, 6, 10, 15, 21, 28, 36, 45, \ldots \rangle}.$$

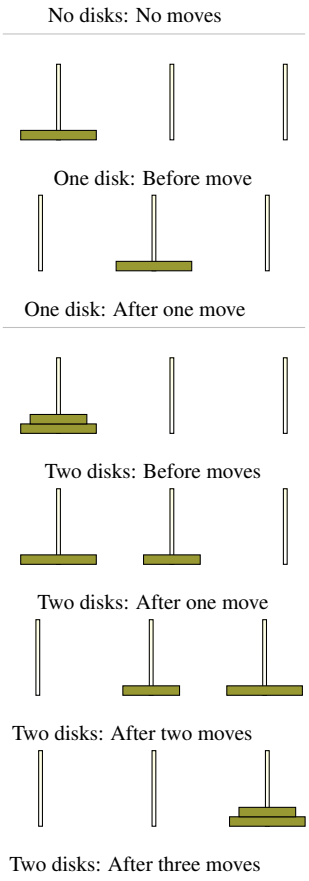
First, the function $t(n)$ satisfies the initial condition

$$t(0) = \frac{0(0-1)}{2} = 0 = t_0$$

Next, by a renaming of variables

$$(t(n) = n(n-1)/2) \Rightarrow \left(t(n-1) = \frac{(n-1)(n-2)}{2} \right)$$

Substitute these expressions into equation 16 and see if it is `True`. Let's substitute $t(n-1) = (n-1)(n-2)/2$ into the triangular equation and follow



the algebra.

$$\begin{aligned} t(n) &= t(n-1) + (n-1) \\ &= \frac{(n-1)(n-2)}{2} + (n-1) \\ &= \frac{(n-1)(n-2) + 2n - 2}{2} \\ &= \frac{n^2 - n}{2} \\ &= \frac{n(n-1)}{2} \end{aligned}$$

The Mersenne sequence & the Tower of Hanoi

The function $m(n) = 2^n - 1$ satisfies the Mersenne recurrence

$$m_n = 2m_{n-1} + 1 \quad \text{for } n > 0, \text{ and initial condition } m_0 = 0.$$

and generates the Mersenne sequence $\vec{M} = \langle 0, 1, 3, 7, 15, 31, 63, 127, 255, 511, \ldots \rangle$.

First, the function $m(n) = 2^n - 1$ satisfies the initial condition

$$m(0) = 2^0 - 1 = 0 = m_0 \quad \text{at } n = 0.$$

The algorithmic idea behind the Mersenne recurrence is: To solve a problem of size n you must solve two similar problems of size $(n-1)$ and perform one extra step.

Next, if $m(n-1) = 2^{n-1} - 1 = m_{n-1}$ for some $n \geq 1$, then substituting $m(n-1) = m_{n-1}$ into the Mersenne equation gives

$$\begin{aligned} m(n) &= 2m(n-1) + 1 \\ &= 2(2^{n-1} - 1) + 1 \\ &= 2^n - 2 + 1 \\ &= 2^n - 1 \end{aligned}$$

Some problems are just fanciful, such as the Tower of Hanoi: Somewhere in southeast Asia, near Hanoi, there is a sect of monks who are moving 64 golden disks from one diamond needle onto another diamond needle subject to God's laws:

1. Move only one disk at a time.
2. Never place a large disk on a smaller one.

God set the orderly, stacked golden disks on one needle and constructed two other diamond needle, which the monks would need. God decreed that when the monks complete their task, the *world will end*.

Speedy monks!

If the monks can move 1 disk every second, you can compute when the world will end by counting the total number of moves they must be made. Rather than trying to compute the number of moves for 64 disks all at once, start by counting the moves for small numbers. A pattern emerges. And, you can establish the pattern is correct using mathematical induction.

As God instructed, it is useful to name things: Let n be the number of disks, and let m_n be the number of moves the monks will need to take. For small values of n, the move count can be computed by examining them all.

Tower of Hanoi: Disks versus moves

Number of disks n	0	1	2	3	4	5	6
Number of moves $m(n)$	0	1	3	7	15	31	63

The trick is to discover the recurrent pattern that *models* the data.

What the monks cannot do is move the bottom disk until the top $(n-1)$ are moved off of it. Once the top disks are moved off, the monks can move the bottom disk. Then, they can move the top $(n-1)$ disks back on top of the bottom disk.

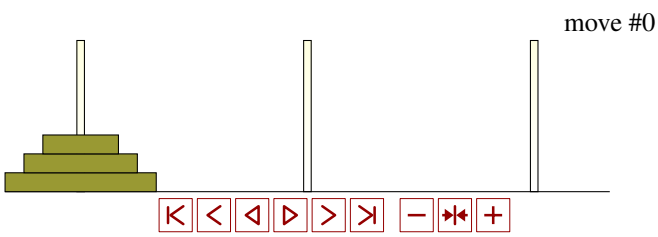
Play the animation below to convince yourself that for 3 disks the top 2 disks must be moved twice (off-of and on-to the bottom disk) and the bottom disk is moved only once. The total number of moves for 3 disks is

$$3 + 1 + 3 = 7$$

The Mersenne equation describes this general relationship.

$$m_n = m_{n-1} + 1 + m_{n-1} = 2m_{n-1} + 1$$

with initial condition $m_0 = 0$



Oh yes, when will the world end? To move 64 disks from one diamond needle to another requires

$$m(64) = 2^{64} - 1 \quad \text{total moves}$$

How long will this take? Let's assume monks can move 1 disk every second. As you know, there are about π billion seconds in a century.

$$\begin{aligned}
1 \text{ century} &= 100 \text{ years} \\
&\approx 36,500 \text{ days} \\
&= 36,500 \text{ days} \times 24 \text{ hours/day} = 876,000 \text{ hours} \\
&= 876,000 \text{ hours} \times 60 \text{ minutes/hour} = 52,560,000 \text{ minutes} \\
&= 52,560,000 \text{ minutes} \times 60 \text{ seconds/minute} \\
&= 3,153,600,000 \text{ seconds} \\
&\approx \pi \text{ billion seconds}
\end{aligned}$$

Use the approximation

$$2^{10} = 1,024 \approx 1,000 = 10^3$$

to make a back-of-the-envelope calculation. The monks can complete their task in

$$\begin{aligned} 2^{64} - 1 \text{ moves} &= 2^{64} - 1 \text{ seconds} \\ &\approx (2^{10})^{6.4} \text{ seconds} \\ &\approx (10^3)^{6.4} \text{ seconds} \\ &\approx 10^{19} \text{ seconds} \\ &= 10 \cdot 10^9 \cdot 10^9 \text{ seconds} \\ &\approx 3\pi \cdot 10^9 \cdot 10^9 \text{ seconds} \end{aligned}$$

That is, the monks will finish in about

$$3\pi \text{ billion billion seconds} \quad \text{or} \quad 3 \text{ billion centuries}$$

22×0.1375 billion centuries ≈ 3 billion centuries

Scientists estimate the big bang occurred about 13.75 billion years ago. That's about 0.1375 billion centuries. Therefore, the monks will finish their task when the known universe is about 22 times its current age!

The approximations used to turn moves into centuries is rather gross: It is off by a factor of about two. 2^{64} seconds is really about 5.846 billion centuries or about 42 times the current estimate for when the big bang occurred!

Annunities

An annuity is a regular series of deposits or payments into an investment or to pay a debt. Pretend you make a fixed payment P each period (week, month, year, your choice) for a total of n periods.

A fixed-rate, 30 home mortgage is the classic example on an annuity. Mortgage payments are made at the beginning of each month for 30 years (360 payments). The monthly interest r does not vary.

The value of annuity increases each period by a fixed rate r. If the value is V at the start of one period, the value will be $P + V + rV = P + (1 + r)V$ at the start of the next period. The three terms are the *regular payment* P, the *previous value* V, and the *interest earned* rV.

That is, the value of the annuity satisfies the recurrence equation

$$V_n = P + (1 + r)V_{n-1} \quad \text{with initial condition } V_0 = P.$$

This equation is very much like the Mersenne equation $m_n = 2m_{n-1} + 1$ except the factor is $(1 + r)$ and not 2, and the shift is P and not 1.

If you follow the first few iterations you can guess the general formula for V_n.

$$\begin{aligned} V_0 &= P \\ V_1 &= P + (1 + r)V_0 = P[1 + (1 + r)] \\ V_2 &= P + (1 + r)V_1 = P[1 + (1 + r) + (1 + r)^2] \\ &\vdots \\ V_n &= P[1 + (1 + r) + (1 + r)^2 + \cdots + (1 + r)^n] \end{aligned}$$

That is, the value V_n is a geometric sum. By theorem 26, its value can be computed by the function

$$V(n) = P\left[\frac{(1+r)^{n+1}-1}{(1+r)-1}\right] = P\left[\frac{(1+r)^{n+1}-1}{r}\right]$$

Consider what happens when you make payments $P = \$1000$ each month for 30 years at a rate $r = 6\%/12 = 0.5\% = 0.005$. The future value of your payments is

$$\begin{aligned} V(360) &= 1000\frac{(1.005)^{361}-1}{0.005} \\ &\approx 20000\cdots 5.05268808831 \\ &\approx 1,010,537.62 \end{aligned}$$

You've forgone the use of \$360,000 over the 30 years, but on the other hand you've almost tripled your savings.

Fibonacci Numbers & Rabbit Populations

Overrun by rabbits, Fibonacci determined to model their growth. Fibonacci modeled the problem by letting Let F_n count the number of rabbit pairs at the start of month n. He assumed:

By the way, rabbits never die.

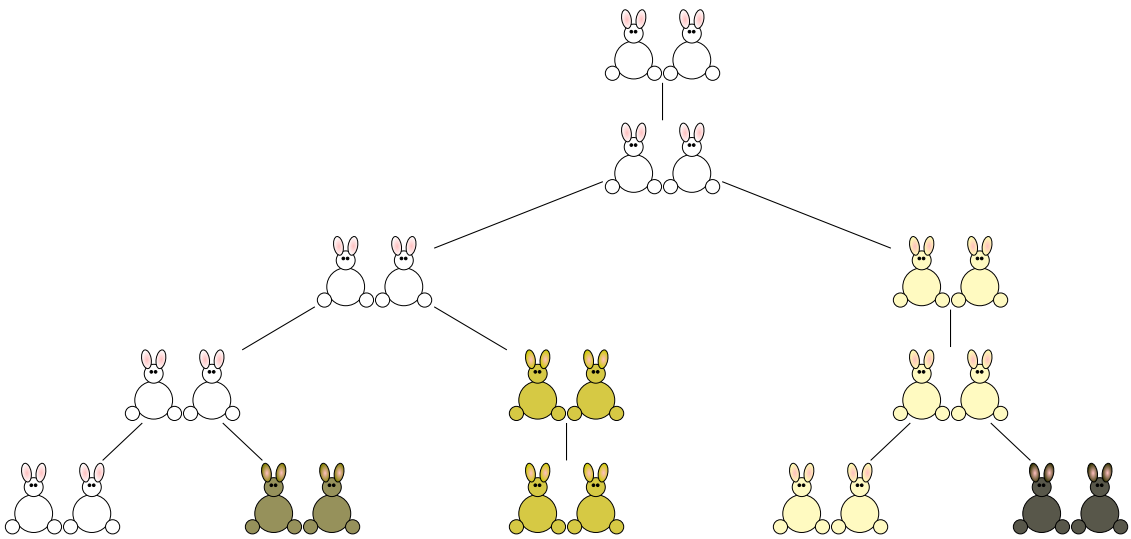
1. In the beginning there were no rabbits: $F_0 = 0$.
2. At the start of rabbit time, God created an original pair of rabbits: $F_1 = 1$.
3. After one month the pair mated: $F_2 = 1$.
4. At the start of the next month they produced a pair of offspring: $F_3 = 2$
5. Each pair repeated the same cycle: At month n every pair of rabbits alive in the previous month remains alive, and every pair alive for 2 or more months breeds another pair of rabbits.

F_{n-1} is the number of pairs during month $(n-1)$. F_{n-2} is the number of pairs old enough to breed another pair. The sum $F_{n-1} + F_{n-2}$ is the number of pairs during month n. The recurrence that describes this model is:

$$F_n = F_{n-1} + F_{n-2}$$

Propagation of rabbits

Rabbit time	0	1	2	3	4	5	6
Original pair		1	1	1	1	1	1
First Generation				1	1	1	1
Second Generation					1	1	1
Third Generation						2	2
Fourth Generation							3
Totals	0	1	1	2	3	5	8



The sequence that counts the number of pairs of rabbits during month n is called Fibonacci sequence. Fibonacci numbers can be computed by the recurrence

$$\boxed{F_n = F_{n-1} + F_{n-2}} \tag{17}$$

with initial condition $F_0 = 0$ and $F_1 = 1$. Terms in the Fibonacci sequence can be computed by the function

$$F(n) = \frac{\varphi^n - \overline{\varphi}^n}{\sqrt{5}}$$

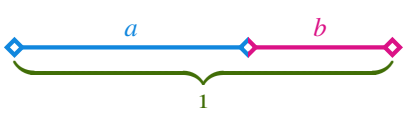
where

$$\varphi = (1 + \sqrt{5})/2 \approx 1.618033 \quad \text{is the golden ratio.}$$

and

$$\overline{\varphi} = \frac{1 - \sqrt{5}}{2} \approx -0.618033 \quad \text{is its conjugate}$$

The golden ratio is often described in terms of ratios of a unit line segment.



1 is to a as a is to b

The golden ratio can also be described in term of a unit area golden rectangle. In the diagram below, the area of the unit rectangle is $a(a+b) = 1$. This area is equal to the sum of two subareas: a^2 and ab. That is,

$$\boxed{\begin{aligned} 1 &= a(a+b) \\ &= a^2 + ab \\ \rightarrow a &= \frac{-b \pm \sqrt{b^2+4}}{2} \end{aligned}}$$

Fibonacci is a *stronger* recurrence: It requires an assignment of values to two initial condition: $F_0 = 0$ and $F_1 = 1$.

φ and $\overline{\varphi}$ are the *zeros* (or *roots*) of the polynomial equation

$$x^2 - x - 1 = 0$$

In particular,

$$\varphi^2 = \varphi + 1$$

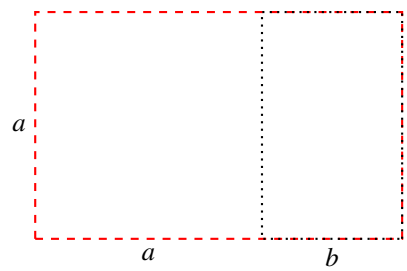
and

$$\overline{\varphi}^2 = \overline{\varphi} + 1$$

Oh! Finding the function that satisfies an equation has something to do with solving polynomial equations.

$$\frac{1}{a} = \frac{a}{b}$$

and $a + b = 1$ implies $a^2 - a - 1 = 0$.



When $b = 1$, the area (height $\times$ width) of the golden rectangle is

$$(a \times (a + b) = (-\hat{\varphi} \times \varphi) \approx (0.618 \times 1.618)$$

The proof that $F(n)$ satisfies the recurrence goes like this:

The Fibonacci recurrence is "second order." That's why two initial conditions are required.

1. The basis must be established for $n = 0$ and $n = 1$. $F_0 = 0$ and $F_1 = 1$ are given. At $n = 0$ the function has value 0

$$F(0) = \frac{\varphi^0 - \overline{\varphi}^0}{\sqrt{5}} = 0$$

At $n = 1$ the function has value 1

$$F(1) = \frac{\varphi^1 - \overline{\varphi}^1}{\sqrt{5}} = \frac{(1 + \sqrt{5})/2 - (1 - \sqrt{5})/2}{\sqrt{5}} = 1$$

2. The induction hypothesis has two parts: For some $n \geq 2$

$$F_{n-1} = \frac{\varphi^{n-1} - \overline{\varphi}^{n-1}}{\sqrt{5}} \quad \text{and} \quad F_{n-2} = \frac{\varphi^{n-2} - \overline{\varphi}^{n-2}}{\sqrt{5}}$$

Recall,

$$\varphi^2 = \varphi + 1$$

and

$$\overline{\varphi}^2 = \overline{\varphi} + 1$$

3. Then the conclusion comes from the calculation below.

$$\begin{aligned} F_n &= F_{n-1} + F_{n-2} \\ &= \frac{\varphi^{n-1} - \overline{\varphi}^{n-1}}{\sqrt{5}} + \frac{\varphi^{n-2} - \overline{\varphi}^{n-2}}{\sqrt{5}} \\ &= \frac{\varphi^{n-2}(\varphi + 1) - \overline{\varphi}^{n-2}(\overline{\varphi} + 1)}{\sqrt{5}} \\ &= \frac{\varphi^{n-2}(\varphi^2) - \overline{\varphi}^{n-2}(\overline{\varphi}^2)}{\sqrt{5}} \\ &= \frac{\varphi^n - \overline{\varphi}^n}{\sqrt{5}} \end{aligned}$$

Time complexity

You've decided it would be worthwhile to monitor the code you've written. Some of it is quick, but some routines bog down when the amount of data to process becomes large. You would like to know the answer to the question: What portions of the code take the most time? The Pareto principle predicts 20% of your code will take 80% of the time.

You may have noticed that there are several ways to solve a problem, yet some seem more efficient than others. You'd like to know why.

Linear complexity

To understand complexity, start with loops. A `for` statement that loops n times and performs one instruction per loop has complexity $T(n) = n$.

```
for (k = 0; k < n; k++) { perform one instruction }
```

The count of operations, the *time complexity*, is the sum of terms in the Alice sequence.

$$n = \sum_{k=0}^{n-1} 1$$

For instance, the worst case time complexity to search a list of size n for a value x takes can take up to n comparisons.

In the best case, linear search takes one operation; and on average under reasonable assumptions, linear search takes $n/2$ operations.

The worst case recursion is: The time to search through n objects is the time to search through $n-1$ times plus the time to perform another test for equality. The time complexity $T(n)$, for th single loop with a fixed number of operations per loop is described by the recurrence equation.

Notice that equation 18 could be written as

$$T(n) = T(n-1) + a\binom{n-1}{0}$$

$$\boxed{T(n) = T(n-1) + a, \quad T(0) = 0} \tag{18}$$

The linear function that solves equation 18 with initial condition $T(0) = 0$ is

The integral of a constant is a linear polynomial. The sum of a constant is a linear expression.

$$T(n) = an$$

An algorithm whose run time grows no faster than some linear function of the input size n has time complexity big O of n, written $O(n)$.

Quadratic complexity

Doubly-nested loops have, in general, quadratic time complexity given by Triangular numbers.

```
for (k = 0; k < n; k++) {
  for (j = 0; j < k; j++) { perform one instruction }
}
```

$$n(n-1)/2 = \sum_{k=0}^{n-1}\left(\sum_{j=0}^{k-1} 1\right) = \sum_{k=0}^{n-1} k$$

For instance, several sorting algorithms use double loops to place objects in order.

The time complexity $T(n)$, for a doubly-nested loop is described by the recurrence equation.

Equation 19 defines a recursion for adding up the natural numbers $0+1+2+\cdots+(n-1)$: The total is the triangular number $n(n-1)/2$. Equation 19 could be written as

$$T(n) = T(n-1) + \binom{n-1}{1}$$

$$\boxed{T(n) = T(n-1) + (n-1)} \tag{19}$$

The function that solves equation 19 with initial condition $T(0) = 0$ is

$$\boxed{T(n) = \frac{n(n-1)}{2} = \binom{n}{2}}$$

An algorithm whose run time grows no faster than some quadratic function of the input size n has time complexity big O of n^2, written $O(n^2)$.

Triply-nested loops occur too: For example, several matrix algorithms require three nested loops. For instance,

$$\boxed{P(n) = P(n-1) + \frac{(n-1)(n-2)}{2}} \tag{20}$$

might describe the complexity of a triply-nested loop. The cubic polynomial that solves equation 20 with initial condition $P(0) = 0$ is

$$\boxed{P(n) = \frac{n(n-1)(n-2)}{6} = \binom{n}{6}}$$

Equation 20 defines a recursion for summing the triangular numbers $0 + 0 + 1 + 3 + 6 + \cdots + n(n-1)/2$: The total is a *pyramidal* number $n(n-1)(n-2)/6$. Equation 20 could be written as

$$P(n) = P(n-1) + \binom{n-1}{2}$$

An algorithm whose run time grows no faster than some polynomial function of degree d has time complexity big O of n^d, written $O(n^d)$. Algorithms that have polynomial time complexity are called *tractable*. These problems are accepted as "easy to solve" in a reasonable number of steps.

Don't be fooled: An n^3 algorithm will take a long time even for relatively small values of n. Pretend a processor can execute 10^{10} instructions per seconds. Then an $O(n^3)$ algorithms will take

$$P(n) \approx \frac{n^3}{10^{10}} \text{ seconds.}$$

For instance when $n = 10^3$ the algorithm will take about $\left(\frac{1}{10}\right)^{\text{th}}$ of a second. How many instructions are needed when $n = 10^6$?. Express your answer in centuries.

Binary search & logarithms

Binary search is a basic *searching* algorithm. The idea is to search for a *key* in a *sorted* list. By recursively discarding one-half of the list until the key is found, or the list becomes empty. The recurrence equation that describes this is

$$b_n = b_{n/2} + 1 \quad \text{for } n = 2, 4, 8, \ldots$$

Phone books are an overused example of binary search. When searching for someone by name, look in the middle page and either: Find what you are looking for, or search the half in which the name could lie based on alphabetical order. Does anyone really look-up names this way? Most likely not. People use other heuristics.

The code is From Rosetta Code

The HASKELL function `binarySearch` acts on a value `p`, on which arithmetic can be performed, and a pair of values `(low, high)` and returns either `\Nothing` or `Just p`.

There are two terminating conditions that stop the recursion are

1. When `high < low` the list is empty. The function returns `\Nothing`. This says `p` was not found between `low` and `high`.

2. When `p` and `mid` are `EQ`. The function returns `Just mid` to denote `p` was found between `low` and `high`.

```
binarySearch :: Integral a => (a -> Ordering) -> (a, a) -> Maybe a
binarySearch p (low,high)
| high < low = \Nothing
| otherwise =
  let mid = (low + high) `div` 2 in
  case p mid of
    LT -> binarySearch p (low, mid-1)
    GT -> binarySearch p (mid+1, high)
    EQ -> Just mid
```

The logarithmic transformation $k = \lg n$ and its inverse $n = 2^k$, allows the renaming

$$b(n) = b(2^k) = g(k) \quad \text{and} b(n/2) = b(2^{k-1}) = g(k-1)$$

This transform the equation

$$b_n = b_{n/2} + 1 \quad \text{for } n = 2, 4, 8, \ldots.$$

to Gauss equation.

$$g_k = g_{k-1} + 1 \quad \text{for } k = 1, 2. 3, \ldots,$$

which you know has solution $g(k) = k$.

Therefore, the *time complexity* is described by

$$k = g(k) = b(n) = \lg n$$

That is, the function $b(n) = \lg n$ satisfies the recurrence

$$b_n = b_{n/2} + 1 \quad \text{for } n = 2, 4, 8, \ldots.$$

Binary search has time complexity big O of gn, written $O(\lg n)$.

Perhaps it is more simple to let $b(n) = \lg n$ so that $b(n/2) = \lg n/2 = \lg n - 1$. The substituting in the equations

$$\begin{aligned} b_n &= b_{n/2} + 1 \\ &= (\lg n - 1) + 1 \\ &= \lg n \end{aligned}$$

Pass a Quiz: Recursive sequences

Take a quiz on page 328 to check your understanding. You can return to here from the quiz.

Newton's root-finding method

Newton devised a fast method to compute *roots* or *zeros* of a function $f(x)$. A root of $f(x)$ is a value of x such that $f(x) = 0$. Newton's method has two step.

1. Make an initial guess that x_0 is a root of $f(x)$

2. Construct a sequence using the recursion

$$\boxed{x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}} \tag{21}$$

   where $f'(x)$ is the derivative of $f(x)$.

One application of Newton's method computes square roots.

The square root of, $\sqrt{2}$, is a zero of the function $f(x) = x^2 - 2$. In this case, $f'(x) = 2x$. Substituting this into equation 21 yields the recurrence equation

$$\begin{aligned} x_n &= x_{n-1} - \frac{x_{n-1}^2 - 2}{2x_{n-1}} \\ &= \frac{2x_{n-1}^2}{2x_{n-1}} - \frac{x_{n-1}^2 - 2}{2x_{n-1}} \\ &= \frac{x_{n-1}^2 + 2}{2x_{n-1}} \\ &= \frac{x_{n-1}}{2} + \frac{1}{x_{n-1}} \\ &= \frac{1}{2}\left(x_{n-1} + \frac{2}{x_{n-1}}\right) \end{aligned}$$

Starting with an initial guess, say $x_0 = 1$, as the square root of 2, the recurrence generates the sequence of approximations

$$\left\langle 1, \frac{3}{2}, \frac{17}{12}, \frac{577}{408}, \ldots \right\rangle$$

or in fixed-point notation

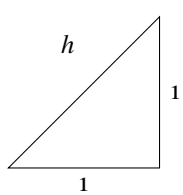
$\sqrt{2} \approx 1.414213562373095\cdots$

$$\langle 1.0,\ 1.5,\ 1.4666\cdots,\ 1.41421568627\cdots, \ldots \rangle$$

Although Newton's name is attached the method, the basic idea was known to ancient mathematicians of Mesopotomia the region of modern day Iraq and Iran. Clay tablets dated from around 1800 B.C. to 1600 B.C. have been found in Mesopotomia that show how to approximate $\sqrt{2}$ and perform other arithmetic operations.

The found artifacts showing computational dexterity date from the time when the Babylonians lived in Mesopotomia.

These early mathematicians considered an isosceles right triangle with legs of length 1



They knew from the Pythagorean theorem theorem that

$$1^2 + 1^2 = h^2$$

so the *hypotenuse* h has length $h = \sqrt{2} \approx 1.41421356\cdots$ The Babylonians knew how to approximate the value of $h = \sqrt{2}$ to many decimals places. They used sexigesimal notation. It is not certain what reasoning the Babylonians used, their calculations indicate this is what they did:

- Let $h_0 = 1$ be an initial approximation to $\sqrt{2}$
- Clearly 1 is too small as the Babylonians could easily measure

- If $1 = \sqrt{2}$, then $1 \cdot 1 = \sqrt{2} \cdot \sqrt{2} = 2$ and $2/1 = \sqrt{2}$
- As it is, $2/1$ is too large
- The average of the under estimate 1 and the over estimate $2/1$ provides a better approximation to $\sqrt{2}$, call this

$$h_1 = \frac{1}{2}\left(h_0 + \frac{2}{h_0}\right) = \frac{1}{2}\left(1 + \frac{2}{1}\right) = \frac{3}{2}$$

- But $h_1 = 3/2$ is too large as the Babylonians could measure
- If $3/2$ were the exact square root, then $2/(3/2) = 4/3$ would equal $\sqrt{2}$
- As it is, $4/3 \approx 1.333\cdots$ is too small
- The average of the over estimate $3/2$ and the under estimate $4/3$ will provide a better approximation

$$h_2 = \frac{1}{2}\left(h_1 + \frac{2}{h_1}\right) = \frac{1}{2}\left(\frac{3}{2} + \frac{4}{3}\right) = \frac{17}{12} \approx 1.41166\cdots$$

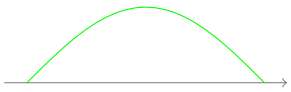
- But $h_2 = 17/12$ is too small
- The Babylonians carried out this iteration a few more times computing the $\sqrt{2}$ accurately to at least 9 decimal places
- That is, they next computed the average of $h_2 = 17/12$ and $2/h_2 = 24/17$

$$\begin{aligned} h_3 &= \frac{1}{2}\left(h_2 + \frac{2}{h_2}\right) \\ &= \frac{1}{2}\left(\frac{17}{12} + \frac{24}{17}\right) \\ &= \frac{1}{2}\left(\frac{289 + 288}{17 \times 12}\right) \\ &\approx 1.41421568628\cdots \end{aligned}$$

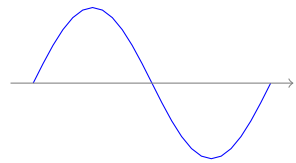
Newton generalized this idea and his method is often a very fast root-finding methods.

Harmonic numbers

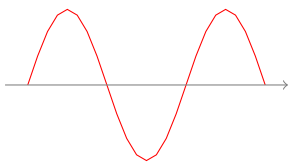
Think of a string, for instance, a piano or guitar string, tied down at boundaries 0 and 1. Pluck the string to set it vibrating. The *fundamental tone* of the string is produced when it is plucked to create a single wave with wavelength $\lambda = 1$, so that there is one extrema: a single maximum. The *second fundamental tone* of the string is produced when it is plucked to create a wave with wavelength $\lambda = 1/2$. Continuing, the *third fundamental tone* of the string is produced when it is plucked to create a wave with three waves each with wavelength $\lambda = 1/3$. These waves are illustrated in the diagram below.



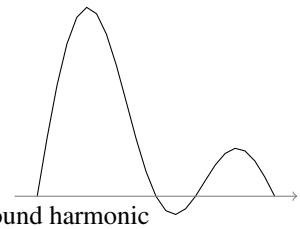
First harmonic: $\sin(x)$



Second harmonic: $\sin(2x)$



Third harmonic: $\sin(3x)$



Compound harmonic

Definition 25: Harmonic sequence

The fractions

$$1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \ldots$$

form the harmonic sequence. *In general,*

$$h_n = \frac{1}{n}, \; n \in \mathbb{Z}^+$$

is the nth number in the harmonic sequence

n	1	2	3	4	5	6	7	8	9	10	...
h_n	1	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{6}$	$\frac{1}{7}$	$\frac{1}{8}$	$\frac{1}{9}$	$\frac{1}{10}$	...

The harmonic sequence is different from most sequences studied in this class: It is a sequence of rational numbers $1/n \in \mathbb{Q}$, $n \in \mathbb{Z}^+$. It is not a sequence of natural numbers.

Although the fundamental tones are of fundamental interest, it is when several fundamental tones of sound (light) are blended that truly interesting events result. Adding the fractions in the harmonic sequence generates the *harmonic numbers*.

Humans can see electromagnetic waves with wavelengths in the range of about 400 to 700 nanometers. We can hear acoustic waves with frequencies (the reciprocal of wavelength) in the range of about 16 Hz to 20 kHz.

Definition 26: Harmonic numbers

The partial sums

$$H_0 = 0, \quad H_1 = 1, \quad H_2 = 1 + \frac{1}{2}, \quad H_3 = 1 + \frac{1}{2} + \frac{1}{3}$$

$$H_4 = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4}, \quad H_5 = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5}, \ldots$$

In general, harmonic numbers satisfy the initial condition $H_0 = 0$ and

recurrence equation

$$H_n = \sum_{k=1}^{n} \frac{1}{k} = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{n}$$
$$= H_{n-1} + \frac{1}{n} \quad n \geq 1$$

Here are the first few Harmonic numbers written in fractional form.

The first few Harmonic numbers

n	1	2	3	4	5	6	7	8	9	10	...
H_n	1	$\frac{3}{2}$	$\frac{11}{6}$	$\frac{25}{12}$	$\frac{137}{60}$	$\frac{49}{20}$	$\frac{363}{170}$	$\frac{761}{280}$	$\frac{7129}{2520}$	$\frac{7381}{2520}$	...

Harmonic recurrences & induction

The recurrence equation for harmonic numbers is

$$H_{n+1} = H_n + \frac{1}{n+1}, \quad \text{for } n \geq 0 \text{ and } H_0 = 0.$$

For example,

$$H_1 = H_0 + \frac{1}{1} = \frac{1}{1}$$
$$H_2 = H_1 + \frac{1}{2} = \frac{3}{2}$$
$$H_3 = H_2 + \frac{1}{3} = \frac{11}{6}$$
$$H_4 = H_3 + \frac{1}{4} = \frac{25}{12}$$

One fact taught in calculus is that the harmonic numbers diverge to ∞.

Theorem 32: Harmonic Numbers diverge to ∞

The $2^{n^{\text{th}}}$ Harmonic number

$$H_{2^n} = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{2^n}$$

is greater than or equal to $1 + \frac{n}{2}$.

$$H_{2^n} \geq 1 + \frac{n}{2}$$

Therefore, $H_n \to \infty$ as $n \to \infty$.

Proof: Harmonic numbers diverge

Notice the terms of the sum are $h_k = 1/k$ *for all* $k = 1, 2, \ldots, 2^n$. *There are three slight changes to what occurs in most induction problems:*

1. *We don't know a simple formula for the sum, but we're claiming it is greater than or equal to the value of the function* $f(n) = 1 + \frac{n}{2}$.
2. *The sum is over the index range from* 1 *to* 2^n.
3. *When* n *is replaced by* $n+1$ *many new terms are added in the summation. These added terms are:*

$$a_{2^n+1} = \frac{1}{2^n+1}, a_{2^n+2} = \frac{1}{2^n+2}, \ldots, a_{2^{n+1}} = \frac{1}{2^{n+}}$$

With these observations, the steps of induction are straightforward.

1. *The basis: for* $n = 0$. *We have*

$$H_{2^0} = \sum_{k=1}^{2^0} \frac{1}{k} = a_1 = 1 \geq 1 + \frac{0}{2}$$

2. *Pretend that*

$$H_{2^n} = \sum_{k=1}^{2^n} \frac{1}{k} \geq 1 + \frac{n}{2} \quad \text{for some } n \geq 0$$

3. *Show that*

$$\sum_{k=1}^{2^{n+1}} \frac{1}{k} \geq 1 + \frac{n+1}{2}$$

Notice that

$$\begin{aligned}
H_{2^{n+1}} &= \sum_{k=1}^{2^{n+1}} \frac{1}{k} = \sum_{k=1}^{2^n} \frac{1}{k} + \sum_{k=2^n+1}^{2^{n+1}} \frac{1}{k} \\
&= H_{2^n} + \sum_{k=2^n+1}^{2^{n+1}} \frac{1}{k} \\
&\geq \left(1 + \frac{n}{2}\right) + \left(\frac{1}{2^n+1} + \frac{1}{2^n+2} + \cdots + \frac{1}{2^{n+1}}\right) \\
&\geq 1 + \frac{n}{2} + 2^n \frac{1}{2^{n+1}} \\
&= 1 + \frac{n}{2} + \frac{1}{2} \\
&= 1 + \frac{n+1}{2}
\end{aligned}$$

Where we have used that there are 2^n *terms in the sum* $\sum_{k=2^n+1}^{2^{n+1}} \frac{1}{k}$ *and each of them is greater or equal to* $\frac{1}{2^{n+1}}$.

Theorem 33: Sum of Harmonic numbers

The sum of the Harmonic numbers up to H_{n-1} is n times $H_n - 1$, that is,

$$\sum_{k=1}^{n-1} H_k = nH_n - n$$

Proof: Sum of Harmonic numbers

1. *The basis: for $n = 1$, we have*

$$\sum_{k=1}^{1-1} H_k = \sum_{k=1}^{0} H_k = 0$$

since the upper index is less than the lower index, that is the sum is empty. Also,

$$\sum_{k=1}^{0} H_k = 1 = H_1 - 1 = 0$$

so the identity is vacuously `True` *(If you don't like vacuous arguments, start the sum at $k = 0$ and recall $H_0 = 0$ by definition, or if you don't like that either, start the induction at $n = 2$ and note that $H_1 = 2H_2 - 2 = 1$)*

2. *The inductive hypothesis: Pretend that*

$$\sum_{k=1}^{n-1} H_k = nH_n - n$$

for some $n \geq 1$.

3. *The inductive step: Show that*

$$\sum_{k=1}^{n} H_k = (n+1)H_n - (n+1)$$

Notice that

$$\begin{aligned}
\sum_{k=0}^{n} H_k &= \sum_{k=0}^{n-1} H_k + H_n \\
&= (nH_n - n) + H_n \\
&= (n+1)H_n - n \\
&= (n+1)\left(H_n + \frac{1}{n+1} - \frac{1}{n+1}\right) - n \\
&= (n+1)H_{n+1} - 1 - n \\
&= (n+1)H_{n+1} - (n+1)
\end{aligned}$$

Asymptotic Approximation of Harmonic numbers

It can be shown that the harmonic number H_n approaches the value of the natural logarithm as n increases. Specifically,

$$H_n = \ln n + \gamma + O(1/2n)$$

where $\gamma \approx 0.5772156649\ldots$ is Euler's constant γ.

Two dimensional recurrence equations

There are two-dimensional recurrences that are interesting and useful too. In the notes on counting subsets, you've seen Pascal's identity for computing the number of subsets of a given size.

- $\binom{n}{k}$ counts k-element subsets of an n-element set.
- $\left[{n \atop k}\right]$ counts k-cycle permutations n objects.
- $\left\{{n \atop k}\right\}$ counts k-subset partitions of an n-element set.

$$\boxed{\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}}$$

In the notes on counting permutations you've seen Stirling's rule of the first kind for computing the number permutations with a given number of cycles.

$$\boxed{\left[{n \atop k}\right] = (n-1)\left[{n-1 \atop k}\right] + \left[{n-1 \atop k-1}\right]}$$

In the notes on counting partitions you've seen Stirling's ruel of the second kind for computing the number partitions of a set into a given number of subsets.

$$\boxed{\left\{{n \atop k}\right\} = k\left\{{n-1 \atop k}\right\} + \left\{{n-1 \atop k-1}\right\}}$$

Homework Questions

Use your time outside of class to solve these problems.

1 Prove the function $T(n) = 3^n - 2$ satisfies the recurrence equation

$$T_n = 3T_{n-1} + 4$$

with initial condition $T_0 = -1$

2 Prove the function $T(n) = \lg(n)$ satisfies the recurrence equation

$$T_{2n} = T_n + 1$$

with initial condition $T_1 = 0$.

3 Use mathematical induction to prove that $s(n) = s_n = 2^n + 3^n$ solves

$$s_n = 5s_{n-1} - 6s_{n-2} \quad \text{with initial conditions } s_0 = 2,\ s_1 = 5.$$

2 and 3 are solutions of the polynomial equation

$$x^2 - 5x + 6 = 0$$

4 Let F_n denote a term in the Fibonacci sequence

n	0	1	2	3	4	5	6	7	8	9	...
F_n	0	1	1	2	3	5	8	13	21	34	...

Use mathematical induction to prove that

$$F_n \leq 2F_{n-1} \quad \text{for } n \geq 2.$$

5 Define the Lucas numbers L_n by changing the initial conditions for the Fibonacci equation.

$$L_n = L_{n-1} + L_{n-2}, n > 1, \quad L_0 = 2,\ L_1 = 1$$

Prove that

$$F_{2n} = F_n L_n$$

where F_{2n} and F_n are Fibonacci numbers.

6 Show that the Lucas numbers L_n satisfy the equation

$$L_n = F_{n+1} + F_{n-1}$$

7 Prove that

$$F_{n+k} = F_k F_{n+1} + F_{k-1} F_n$$

8 Prove that for $n \geq 1$

$$\gcd(F_n,\ F_{n-1}) = 1$$

9 Prove Cassini's identity

$$F_{n+1}F_{n-1} - F_n^2 = (-1)^n, \quad \text{for } n > 0$$

where F_k is a Fibonacci number.

10 Let s and t be strings over some alphabet. Give recursive definitions for the following operatios

10.1 `head(s)` that returns the first character in s, call it s_0.

10.2 `tail(s)` all characters in s after s_0.

10.3 `concat` that joins s and t to form the single string st.

10.4 `length(s)` returns the number of characters in s.

10.5 `reverse(s)` that returns a string r formed by writing the characters of s from back to front. string, that is, define a function $r(s)$ that reverses the characters in string s.

Often, people write s^R to denote the reversal of s.

11 Quicksort can be informally described as follows:

- Place the first element in its correct location (this will take n steps on a list of size n).
- Quicksort the list to the left of where the first element landed.
- Quicksort the list to the right of where the first element landed.

Assume that we were lucky enough to always divide the list into two equal sized lists. What recurrence equation models this?

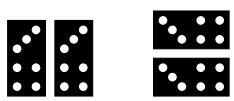
12 Morse code is a series of dots (·) and dashes (—). Pretend a dot can be typed in 1 second while a dash takes 2 seconds to type. Find a recurrence that describes how many messages can be typed in n seconds.

13 Suppose there is always a 1 second pause between dots and dashes in a message. Pretend messages always begin and end with a dot or a dash. What recurrence equation describes the number of messages than can be sent in n seconds?

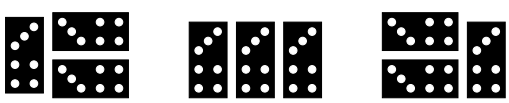
14 In how many ways can $n \geq 1$ be expressed as a sum of 1's and 2's? Different term orders are considered different sums.

15 A domino is a rectangular tile 1 unit long and 2 units high, for instance

Dominoes can be laid out to tile an $n \times 2$ area. For example, there are 2 ways to tile a 2×2 area



And 3 ways to tile a 3×2 area:



Find a recurrence relation that counts the number of $n \times 2$ tilings.

16 Many algorithms have time complexity that can be modeled by the function $q(n) = n \lg n$. Show that $q(n)$ solves the recurrence relation

$$q_n = 2q_{n/2} + n \quad \text{for } n = 2, 4, 8, \ldots$$

with $q_0 = 0$.

17 Let n be a natural number greater than 0. Consider the sequence generated by the recurrence equation

$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{n}{x_k}\right), \quad k \geq 0, \quad x_0 = 1.$$

17.1 Let $n = 1$. What sequence is generated by the recurrence equation?

17.2 Let $n = 2$. To what number does the sequence $\langle 1, x_1, x_2, \ldots \rangle$ converge?

17.3 Let $n = 2$. To what number does the sequence $\langle 1, x_1, x_2, \ldots \rangle$ converge?

18 Write Stirling's approximation from Theorem 30 using natural logarithms.

19 Consider the complexity of chess and Go. For chess, I've read there are 40 moves on average from a given chess position and a game is typically about 50 moves. On the other hand, in the game of go, there are about 250 next positions and a game is typically 250 moves.

Use this information to estimate the time complexity of exploring every possible final state in chess and go. What does this say about the potential difficulty in developing algorithms that successfully play these games?

20 The pizza was delievered uncut. You'll need to cut it. A cut is a straight line from one edge through to another edge: A cut need not go through the center of the pizza.

1. You're expecting only a few friends to share the pizza with you, so you want to cut the pizza into the fewest number of pieces with n cuts.
   (a) List the fewest number of pieces you can make with 0, 1, 2, 3, 4 cuts.
   (b) Find a recurrence relation solution that describes the fewest number of pieces with n cuts.
   (c) Find a closed form solution that describes the fewest number of pieces with n cuts.
2. You're expecting a lot of friends to share the pizza with you, ' so you want to cut the pizza to maximize the number of pieces.
   (a) List the maximum number of pieces you can make with 1, 2, 3, 4 cuts.
   (b) Find a recurrence relation solution that describes the maximum number of pieces with n cuts.
   (c) Find a closed form solution that describes the maximum number of pieces with n cuts.

21 Consider the sequence defined by

$$x_{n+2} + x_n = 0 \quad x_0 = 0,\ x_1 = 1$$

What are the first few terms in the sequence defined by this recurrence? Why might this be called the *sine* sequence?

22 Consider the sequence defined by

$$x_{n+2} + x_n = 0 \quad x_0 = 1,\ x_1 = 0$$

What are the first few terms in the sequence defined by this recurrence? Why might this be called the *cosine* sequence?

Pass a Quiz: Summative exam # 4 on counting and mathematical induction

Take a quiz on page 329 to check your understanding. You can return to here from the quiz.

13. Sorting Data: Permutations & Orders

Permutations

Ineffective Sorts

Let $\mathbb{X}$ be a set. The set of permutations on $\mathbb{X}$ forms a group, called the symmetric group of $\mathbb{X}$. The group operation is function composition.

- A permutation of a permutation is a permutation, called closure.
- Function composition is associative, so permutation actions are associative
- There is an identity permutation that doesn't move anything.
- Every permutation has an inverse that reverses its action.

Symmetric groups play important roles in modern physics.

Recall, from the notes on permutations A permutation is a one-to-one function from a set $\mathbb{X}$ onto itself. Since permutations are about the *order* of things, thinking of a permutation as mapping a sequence to a sequence may be more appropriate.

Also, recall $n!$ permutations can be defined on an n-element set.

A permutation can be written in *cycle notation*. For instance, the permutation $\langle b, c, a\rangle$ of $\langle a, b, c\rangle$ can be written in *cycle notation* as

$$[1, 3, 2]$$

Think of $[1, 3, 2]$ acting on the sequence $\langle a, b, c\rangle$ to produce a permuted

sequence $\langle b, c, a\rangle$.

$$[1, 3, 2]\,\langle a, b, c\rangle = \langle b, c, a\rangle$$

The action of the permutation is:

Move the value a in position 1 to position 3

Move the value b in position 2 to position 1

Move the value c to position 3 to position 2

You have to be careful making these moves: Don't write over a value before you've recorded it.

The sequence $\langle a, b, c\rangle$ can be represented in cycle notation

$$\langle a, b, c\rangle = [1][2][3] \quad \text{the identity permutation.}$$

The inverse of $[1, 3, 2]$ sorts $\langle b, c, a\rangle$ into alphabetical order. The inverse must move the values in positions 1, 2 and 3 to positions 2, 3 and 1.

$$[1, 2, 3]\,\langle b, c, a\rangle = \langle a, b, c\rangle$$

Or, without mention of the values,

$$[1, 2, 3][1, 3, 2] = [1][2][3]$$

The application of the permutations is from right to left and in function composition notation $g \circ f(x) = g(f(x))$.

The composition $[1, 2, 3][1, 3, 2]$ starts on the right and moves 1 to 3, 2 to 1, and 3 to 2. Then $[1, 2, 3]$ is applied moving 3 to 1, 1 to 2, and 2 to 3.

A 2-line notation can help interpret cycle notation. The top line list the objects in their natural order. The bottom line lists where these values are mapped (permuted).

$$\begin{array}{ll} \text{Natural (Input) Order} & 1 \quad 2 \quad 3 \\ \text{Move to Position} & \begin{pmatrix} 3 & 1 & 2 \end{pmatrix} \end{array}$$

$$\begin{aligned} 1 &\mapsto 3 \\ 2 &\mapsto 1 \\ 3 &\mapsto 2 \end{aligned}$$

A third notation uses a permutation matrix.

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

This matrix maps $[1, 2, 3]$ to $[2, 3, 1]$.

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$$

To sort $[2, 3, 1]$, use the inverse matrix

$$P^{-1} = P^T = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

To *sort* is to arrange things based on a criterion or relation. If the relation is reflexive, antisymmetric, and transitive, then the things can be sorted: Ordered. Sorting is the process of computing a permutation that places objects in *order*. For example, to place $\langle c, a, b\rangle$ in alphabetic order, apply the permutation

"move the first letter 'c' to the third position"

"move the second letter 'a' to the first position"

"move the third letter 'b' to the second position"

By this permutation, the input 'c,' 'a,' 'b,' is sorted into the natural order 'a,' 'b,' 'c.'

Orders

The word *order* is used in many ways. There are too many ways in which the word *order* is used for me to count: See Wikipedia's list.

An order could be: A religious order, a rank in the biological taxonomy, what you place with a waiter, a measure of entropy, and an arrangement of objects.

Total orders

Definition 27: Total order

A total or linear order $<$ *satisfies two conditions:*

1. Trichotomy law: *Exactly of the possibilities below is* `True`.

$$a < b, \quad a = b, \quad a > b$$

2. Transitive *law: If* $a < b$ *and* $b < c$, *then* $a < c$.

Essentially, a total order allows the comparison and ordering any two object from the set under consideration. You can compare two numbers, two strings over an alphabet, or any other two object when a total order can be defined.

Example total orders are:

- The English letters: $a < b < c < d < \cdots < z$
- The Natural numbers numbers: $0 < 1 < 2 < 3 < 4 < \cdots$
- The Integers: $\cdots < -3 < -2 < -1 < 0 < 1 < 2 < 3 < \cdots$
- More generally, the Real numbers are totally ordered.

You must be careful when comparing two floating point number.

I like to ask if "less than" ($<$) on the integers $\mathbb{Z}$ is antisymmetric. Many people will say "no" because they "think"

$$(a < b) \wedge (b > a) \Rightarrow a = b$$

cannot be `True`. But, they forget that if the assumption P of a conditional is `False`, then the conditional $\mathrm{P} \Rightarrow \mathrm{Q}$ is `True`, not matter what. In this case,

$$(a < b) \wedge (b > a) \quad \text{is } \texttt{False}$$

The truth value of the conclusion $a = b$ does not matter. The conditional

$$(a < b) \wedge (b > a) \Rightarrow a = b$$

is logically `True`.

Partial orders

There are interesting examples where objects in a set can only be *partially ordered*. That is, there can be values for which the trichotomy law fails: Values a and b that cannot be compared. But, if lesser properties are employed, the values can be partially ordered.

Definition 28: Partial order

A partial order $\leq$ *satisfies three conditions:*

1. Transitive law: *For all a, b and c, if $a \le b$ and $b \le c$, then $a \le c$.*

$$(\forall a, b, c)(((a \le b) \wedge (b \le c)) \Rightarrow (a \le c)$$

2. Reflexive law: *For all a, $a \le a$.*

$$(\forall a)(a \le a)$$

3. Antisymmetric law: *For all a and b, if $a \le b$ and $b \le a$ then $a = b$.*

$$((a \le b) \wedge (b \le a)) \Rightarrow a = b$$

Divides is a partial order on the natural numbers

Divides is an interesting number theoretic idea, which induces a partial order on the natural numbers. An understanding of *divides* is crucial if you want to comprehend modular arithmetic, hashing, cryptography, and many other applications.

Here are some examples of divides.

- 5 divides 35 because $5 \cdot 7 = 35$.
- 8 divides 88 because $8 \cdot 11 = 88$.
- 5 does not divide 37 because 37 divided by 5 leaves a remainder of 2
- 8 does not divide 81 because 81 divided by 8 leaves a remainder of 1

Write

$$a \mid b \quad \text{to express } a \text{ divides } b.$$

and

$$a \nmid b \quad \text{to express } a \text{ does not divide } b.$$

Definition 29: Divides

Let a and b be natural numbers. If there is a natural number c such that

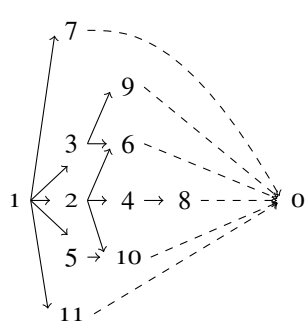
$$ac = b$$

then a divides b. In symbols

$$a \mid b \equiv (\exists c \in \mathbb{N})(ac = b)$$

Here's a graph of *divides* for the first few natural numbers. An edge $a \to b$ means a divides b.

Transitive and reflexive edges are not shown, they can be inferred by context.



Note that 0 divides 0 in number theory. In analysis, where limits are involved, this may not be the case.

The graph shows 1 divides every number, on the other hand, 1 is the only number that divides 1. At first level of the graph are prime numbers: They have exactly two divisors; 1 and themselves. Interestingly, since every number divides 0, it is the "largest" value under this ordering.

Subset is a partial order on a power set

The subset relation is another important partial order. Let $\mathbb{S}$ be a set and let $2^{\mathbb{S}}$ be the power set of $\mathbb{S}$. That is, $2^{\mathbb{S}}$ is the set of all subsets of $\mathbb{S}$. Let $\mathbb{X}$, $\mathbb{Y}$ and $\mathbb{W}$ be elements in $2^{\mathbb{S}}$: $\mathbb{X}$, $\mathbb{Y}$ and $\mathbb{W}$ subsets of $\mathbb{S}$.

- The subset relation is *transitive*

$$(\forall\, \mathbb{X},\, \mathbb{Y},\, \mathbb{W})(((\mathbb{X} \subseteq \mathbb{Y}) \wedge (\mathbb{Y} \subseteq \mathbb{W})) \Rightarrow (\mathbb{X} \subseteq \mathbb{W})) \quad \text{Subset is } \textit{transitive}.)$$

- Every set is a subset of itself, subset is a *reflexive relation*.

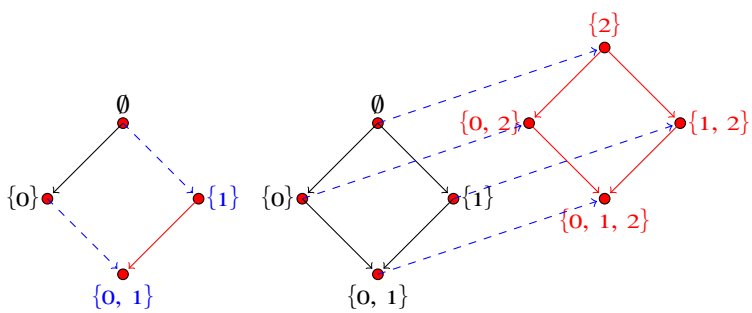
$$(\forall\, \mathbb{X})(\mathbb{X} \subseteq \mathbb{X}) \quad \text{Subset is } \textit{reflexive}.$$

- The subset relation is antisymmetric

$$(\forall\, \mathbb{X},\, \mathbb{Y})(((\mathbb{X} \subseteq \mathbb{Y}) \wedge (\mathbb{Y} \subseteq \mathbb{X})) \Rightarrow (\mathbb{X} = \mathbb{Y})) \quad \text{Subset is antisymmetric.})$$

You can draw graphs showing the partial order induced by the subset relation. Start with the empty set $\emptyset$ which is just a point $\overset{\emptyset}{\bullet}$. The graph for a one element set $\{0\}$ is a line segment $\overset{\emptyset}{\bullet} \longrightarrow \overset{\{0\}}{\bullet}$. With a two element set, the graph is a square, and with three elements you get a a cube.

I will not attempt to draw a hypercube: drawing two cubes and connecting corresponding nodes.



Here's one way to visualize the subset relation on a 4 element set. It's from TEXample.net and by Yury Chebiryak Notice how a 4-bit number identifies a subset of, say, $\{a, b, c, d\}$: 0000 is the empty set; 1111 is the entire set 0101 is the set $\{b, d\}$, and so on. The edges show a Gray code cycle in the 4-cube where vertices along the cycle differ by one bit.

Sorting algorithms

There are many sorting algorithms. The video below from panthema.net shows the action of 15 sorting algorithms (5:48).

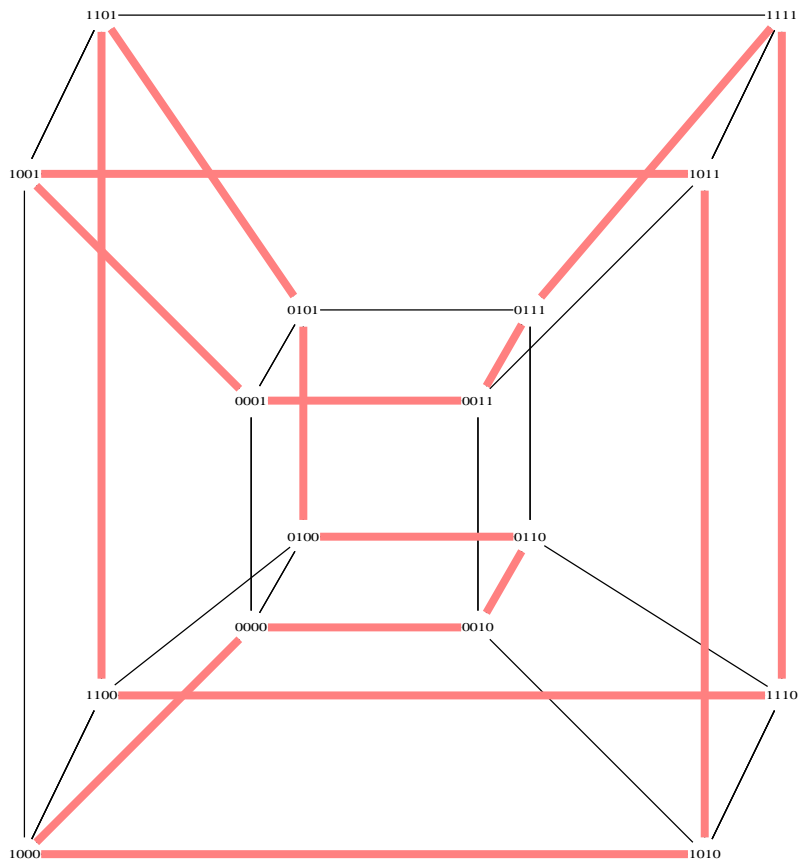
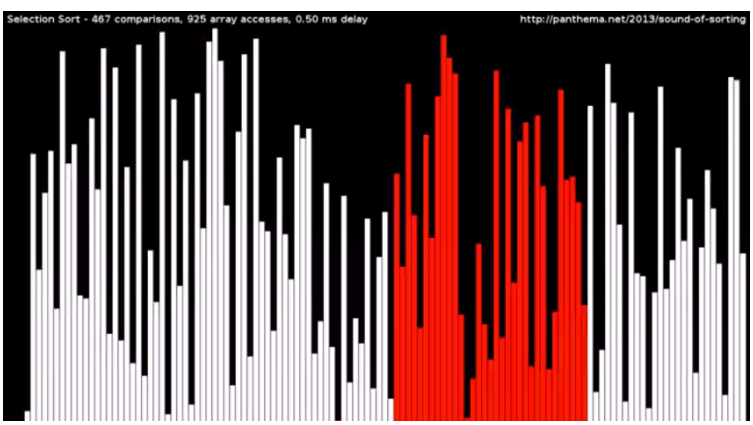


Figure 9: Complete Gray Code



Some sorting algorithms do not require comparisons. Using the properties of the things being sorted, they act faster than $O(n \lg n)$.

There are many qualities one would like in a sorting algorithm. Perhaps the most important is *time complexity*, measured by the number of comparisons. and swaps made when sorting the data. Some sorting algorithms require about n^2 comparisons on average. Others have average case time complexity that have $n \lg n$ behavior.

n^2 *algorithms:* Insertion, selection, and bubble

$$n = 1000 \text{ (one thousand)} \Rightarrow \text{about } 1,000,000 \text{ (one million) steps}$$

$n \lg n$ algorithms: Merge, heap, and quick sort

$$n = 1000 \text{ (one thousand)} \Rightarrow \text{about } 20,000 \text{ (twenty thousand) steps}$$

Pass a Quiz: Orders

Take a quiz on page 330 to check your understanding. You can return to here from the quiz.

Counting relations by property

Recall, from the earlier notes on counting relations there are $2^{|\mathbb{X}|\times|\mathbb{Y}|}$ relations from $\mathbb{X}$ to $\mathbb{Y}$. One way to recognize this is that a relation between $\mathbb{X}$ and $\mathbb{Y}$ can be modeled by an adjacency matrix

A relation is a subset of the Cartesian product

$$\mathbb{X} \times \mathbb{Y} = \{(x, y) : x \in \mathbb{X}, y \in \mathbb{Y}\}$$

If $n = |\mathbb{X}|$ and $m = |\mathbb{Y}|$ are the count of elements in these sets, then $|\mathbb{X} \times \mathbb{Y}| = nm$ and $\mathbb{X} \times \mathbb{Y}$ has 2^{nm} subsets. That is, there are 2^{nm} relations between $\mathbb{X}$ and $\mathbb{Y}$.

		$\lvert\mathbb{Y}\rvert = m$ columns					
		0	1	2	$\cdots$	$(m-2)$	$(m-1)$
	0						
	1						
$\lvert\mathbb{X}\rvert = n$	2			Fill with 0's and 1's			
rows	$\vdots$			in any way you like			
	$(n-2)$						
	$(n-1)$						

And, there are 2^{nm} ways to filled such a matrix.

It is useful to count relations by way the properties they obey: Reflexivity, symmetry, antisymmetry, and transitive.

Counting reflexive relations

a relation $\sim$ is the set of ordered pairs that satisfy the relation. A *reflexive* relation is one where $x \sim x$ for all values of x. And, a relation is an adjacency matrix. A *reflexive* relations is one where there is a 1 at each entry in the diagonal of the matrix.

	a	b	c	d	$\cdots$	u
a	1	x	x	x	$\cdots$	x
b	x	1	x	x	$\cdots$	x
c	x	x	1	x	$\cdots$	x
d	x	x	x	1	$\cdots$	x
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
u	x	x	x	x	$\cdots$	1

The values in off-diagonal entries do not effect reflexivity.

Cardinality n is expressed $|\mathbb{X}| = n$.

Theorem 34: Counting of Reflexive Relations

Let $\mathbb{X}$ be a set with cardinality n. There are $r(n) = 2^{n(n-1)}$ reflexive relations on $\mathbb{X}$.

There are 2^{n^s} relations that can be defined on $\mathbb{X}$.

Proof: Counting reflexive relations

Let A be the adjacency matrix for a reflexive relation $\sim$. There are $n \times n = n^2$ entries in A, and each entries can take on one of 2 values: `True` (1) *or* `False` (1).
To be reflexive, the matrix A must have 1's along its main diagonal. The off-diagonal entries of A can be either 0 or 1. The number of entries in the lower triangle of A is a triangular number

$$1 + 2 + 3 + \cdots + (n-1) = \frac{n(n-1)}{2}$$

For instance, there are $10 = 5(5-1)/2$ entries in the lower triangle of a 5×5 matrix.

	a	b	c	d	e
a	1				
b	7	1			
c	4	8	1		
d	2	5	9	1	
e	1	3	6	10	1

The total number of off-diagonal entries is

$$2 \times \frac{n(n-1)}{2} = n(n-1) = n^2 - n$$

Each off-diagonal entry can have one of two values. Therefore there are

$$r(n) = 2^{n(n-1)} \quad \text{reflexive relations on an } n \text{ element set.}$$

Counting symmetric relations

A *symmetric* relation is one where $(x \sim y) \Rightarrow (y \sim x)$ for all values of x and y. When represented as an adjacency matrix, *symmetry* means that if there is a 1

in row x, column y, then there is a 1 in row y, column x.

	a	b	c	d	$\cdots$	u
a	x	0	1	x	$\cdots$	0
b	0	x	0	x	$\cdots$	0
c	1	x	x	1	$\cdots$	0
d	x	0	1	x	$\cdots$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
u	0	0	1	0	$\cdots$	x

In this case the lower triangle completely determines the upper triangle, and values along the diagonal are unconstrained.

There are

$$1 + 2 + 3 + \cdots + (n-1) + n = \frac{n(n+1)}{2}$$

, entries on or below the main diagonal. Therefore there are

$$sr(n) = \sqrt{2^{n(n+1)}} \quad \text{symmetric relations on an } n \text{ element set.}$$

Notice that

$$2^1 \cdot 2^2 \cdot 2^3 \cdots 2^{(n-1)} \cdot 2^n = 2^{\frac{n(n+1)}{2}}$$

Theorem 35: Counting of Symmetric Relations

Let $\mathbb{X}$ be a set with cardinality n. There are $s(n) = \sqrt{2^{n(n+1)}}$ symmetric relations that can be defined on $\mathbb{X}$.

Counting antisymmetric relations

Now consider the adjacency matrix A for an antisymmetric relation. An antisymmetric relation is constrained by the rule:

$$\text{If } i \neq j, \text{ then } A(i,\ j) = 1 \text{ implies } A(j,\ i) = 0.$$

Theorem 36: Count of Antisymmetric Relations

Let $\mathbb{X}$ be a set with cardinality n. There are

$$a(n) = 2^n 3^{n(n-1)/2} = \left(2\sqrt{3^{n-1}}\right)^n$$

symmetric relations on $\mathbb{X}$.

Proof: Counting antisymmetric relations

To be antisymmetric, the matrix A can not have 1's in symmetric off-diagonal entries: if 1 is in row i, column j then 0 is in row j, column i, unless $i = j$.

There are $n(n-1)/2$ entries in the lower triangle of A. There are 3 possible assignments to each lower triangle entry $a(i, j)$, $i > j$ and its symmetric partner $a(j, i)$.

1. $a(i, j) = 0$ and $a(j, i) = 0$
2. $a(i, j) = 0$ and $a(j, i) = 1$
3. $a(i, j) = 1$ and $a(j, i) = 0$

Therefore, there are $3^{n(n-1)/2}$ ways to fill off-diagonal entries in an antisymmetric matrix A.
To completely fill in A, the diagonal entries must be set to 0 or 1. It does not matter which value is chosen, because the diagonal is not involved in antisymmetry. The diagonal can be filled in 2^n ways. Therefore there are

$$a(n) = 2^n 3^{n(n-1)/2} = \left(2\sqrt{3^{n-1}}\right)^n$$

antisymmetric relations on an n element set.

Counting transitive relations

There is no simple formula $t(n)$ that counts the number of transitive relations on an n element set $\mathbb{X}$. However, these counts can be computed. The Online Encyclopedia of Integer Sequences details this sequence of counts transitive relations. The first few terms in the sequence are

$$\langle 1, 2, 13, 171, 3994, 154303, 9415189, 878222530, 122207703623, 24890747921947, \ldots \rangle$$

There is 1 transitive relation on the empty set $\emptyset$. It is the empty relation.

Both matrices $[0]$ and $[1]$ represent a transitive relation on a 1-element set $\{0\}$.

You can verify that each of the 13 adjacency matrices below represents a transitive relation.

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \tag{22}$$

$$\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Counting partial orders

There is no simple formula $p(n)$ that counts the number of partial orders on an n element set $\mathbb{X}$. However, these counts can be computed. The Online Encyclopedia of Integer Sequences details this sequence of counts partial orders. The first few terms in the sequence are

$$\langle 1,\ 1,\ 3,\ 19,\ 219,\ 4231,\ 130023,\ 6129859,\ 431723379, \ldots \rangle$$

There is 1 partial order on the empty set $\emptyset$. It is the empty relation.

Only $[1]$ is the adjacency matrix representation of a partial order on a 1-element set $\{0\}$.

You can verify that each of the 3 adjacency matrices below represents partial orders.

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \tag{23}$$

Pass a Quiz: Counting orders by their properties

Take a quiz on page 330 to check your understanding. You can return to here from the quiz.

Counting permutation by cycles

Recall, from the earlier notes on permutations there are $n!$ permutations on an n-element set. Also, recall a permutation P can be represented in *cycle notation*.

For instance, the permutation $\langle a,\ c,\ d,\ b \rangle$ of the natural order of these letters can be written in cycle notation as

$$[a][b,\ d,\ c]$$

Read $[a][b,\ d,\ c]$ as saying a goes to a's place; b goes to d's place; d goes to c's place; c goes to b's place. The permutation matrix is

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Stirling's numbers of the first kind count the number of k-cycle permutations of n objects.

The first figure below show the 6 permutations of $\langle 0,\ 1,\ 2 \rangle$ organized by the number of cycles in the permutation: 1-cycle, 2-cycles, or 3-cycles. The first figure below show the 24 permutations of $\langle 0,\ 1,\ 2,\ 3 \rangle$ organized by the number of cycles in the permutation: 1-cycle, 2-cycles, 3-cycles, or 4-cycles. The permutations are color coded. Notice how each permutation of $\langle 0,\ 1,\ 2 \rangle$ generates 4 permutations of $\langle 0,\ 1,\ 2,\ 3 \rangle$.

The permutations on $\{0,\ 1,\ 2,\ 3\}$ can be defined recursively from the permutations on $\{0,\ 1,\ 2\}$. That is, for instance, to build all 2-cycle permutations of $\{0,\ 1,\ 2,\ 3\}$. use the 1 and 2-cycle permutations of $\{0,\ 1,\ 2\}$.

1. Append the cycle $[3]$ to each 1-cycle permutation of $\{0,\ 1,\ 2\}$

$$[0,\ 1,\ 2] \mapsto [0,\ 1,\ 2][3]$$
$$[0,\ 2,\ 1] \mapsto [0,\ 2,\ 1][3]$$

The $6 = 3!$ Permutations on 3 things by cycle

1-cycle	2-cycles	3-cycles
[0, 1, 2]	[0][1, 2]	[0][1][2]
[0, 2, 1]	[1][0, 2]	
	[2][0, 1]	

Figure 10: Cyclic notation for the $3! = 6$ permutations of $\{0, 1, 2\}$.

The $4! = 24$ Permutations on 4 things by cycle

1-cycle	2-cycles	3-cycles	4-cycles
[0, 1, 2, 3]	[0, 1, 2][3]	[0][1, 2][3]	[0][1][2][3]
[0, 1, 3, 2]	[0, 2, 1][3]	[1][0, 2][3]	
[0, 3, 1, 2]	[0][1, 2, 3]	[2][0, 1][3]	
[0, 2, 1, 3]	[0][1, 3, 2]	[0][1][2, 3]	
[0, 2, 3, 1]	[0, 3][1, 2]	[0][1, 3][2]	
[0, 3, 2, 1]	[1][0, 2, 3]	[0, 3][1][2]	
	[1][0, 3, 2]		
	[1, 3][0, 2]		
	[2][0, 1, 3]		
	[2][0, 3, 1]		
	[2, 3][0, 1]		

Figure 11: Cyclic notation for the $4! = 24$ permutations of $\{0, 1, 2, 3\}$.

2. Insert new element 3 in three positions in each 2-cycle permutations of $\{0, 1, 2\}$

$$[0, 1][2] \mapsto [0, 1][2, 3]$$
$$[0, 1][2] \mapsto [0, 1, 3][2]$$
$$[0, 1][2] \mapsto [0, 3, 1][2]$$

$$[0, 2][1] \mapsto [0, 2][1, 3]$$
$$[0, 2][1] \mapsto [0, 2, 3][1]$$
$$[0, 2][1] \mapsto [0, 3, 2][1]$$

$$[1, 2][0] \mapsto [1, 2][0, 3]$$
$$[1, 2][0] \mapsto [1, 2, 3][0]$$
$$[1, 2][0] \mapsto [1, 3, 2][0]$$

Using $\left[{4 \atop 2}\right]$ to name the count of 2-cycle permutations of a 4-element set, write

$$\begin{bmatrix} 4 \\ 2 \end{bmatrix} = 3\begin{bmatrix} 3 \\ 2 \end{bmatrix} + \begin{bmatrix} 3 \\ 1 \end{bmatrix} = 3 \cdot 3 + 2 = 11$$

These eleven permutations are shown in figure 11.

Recall the theorem about Stirling's numbers of the first kind. The symbol $\left[{n \atop k}\right]$ counts the number k-cycle permutations on n things. Stirling numbers of the first kind satisfy the boundary conditions

$$\begin{bmatrix} n \\ 0 \end{bmatrix} = 0, \ (n = 1, \ldots) \qquad \begin{bmatrix} n \\ n \end{bmatrix} = 1, \ (n = 0, 1, 2, \ldots)$$

and the two-dimensional recurrence equation

$$\begin{bmatrix} n \\ k \end{bmatrix} = (n-1)\begin{bmatrix} n-1 \\ k \end{bmatrix} + \begin{bmatrix} n-1 \\ k-1 \end{bmatrix}$$

Homework Questions

Use your time outside of class to solve these problems.

1 Recall, the listing of relations on a two-element set as adjacency matrices on page 181: There are $2^{2 \cdot 2} = 2^4 = 16$

1.1 Use Theorem 34 to determine how many are reflexive. Identify them from the list.

1.2 Use Theorem 35 to determine how many are symmetric. Identify them from the list.

1.3 Use Theorem 36 to determine how many are antisymmetric. Identify them from the list.

1.4 Which of these adjacency matrices represent partial orders?

2 "Please Excuse My Dear Aunt Sally" is a mnemonic to help people remember the precedence order to arithmetic operations. What does PEMDAS mean?

3 Programming languages support numerous operations: Arithmetic, comparison, logical, and other categories. For your favorite programming language, what are the basic categories and how is precedence (order of operation) defined. If you can't choose a programming language use C or HASKELL.

Rock-paper-scissors-lizard-Spock was invented by Sam Kass with Karen Bryla. The game was mentioned several times on the Big Bang Theory.

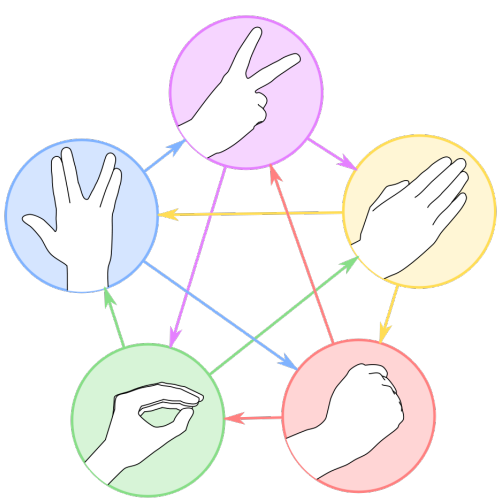
4 Packet switching is a network communication idea where a long message is transmitted as a collection of shorter messages called packets. Packets can arrive at their destination in almost any order.

Pretend a sequence of packets arrives in the order $\langle 4, 2, 1, 0, 3 \rangle$. What permutation will sort the packets into their sorted order? Write your answer using cyclic notation.

5 Consider the rules for Rock–Paper–Scissors–Lizard–Spock where "*x beats y*" is given colorful names.

- Scissors cuts paper
- Paper covers rock
- Rock crushes lizard
- Lizard poisons Spock
- Spock smashes scissors
- Scissors decapitates lizard
- Lizard eats paper
- Paper disproves Spock
- Spock vaporizes rock
- Rock crushes scissors

The graph below shows what beats what.



Is the game transitive? Is it antisymmetric?

6 Verify that the three adjacency matrices not listed in 22 above represent relations that are not transitive.

7 Verify that the three adjacency matrices listed in 23 above represent partial orders.

8 Show that it is not possible to have a non-trivial cycle in a partial order. That is, there cannot exist different elements a, b, and c such that $a \preceq b$ and $b \preceq c$ and $c \preceq a$.

9 CANFOOL(p, t) is a relation between people p and q (see problem 4 from the homework questions on logic for control). Ambiguity is common about people. Answer these questions from your point of view.

9.1 Is CANFOOL(p, t) reflexive? What does your answer mean?

9.2 Is CANFOOL(p, t) symmetric? What does your answer mean?

9.3 Is CANFOOL(p, t) antisymmetric? What does your answer mean?

9.4 Is CANFOOL(p, t) transitive? What does your answer mean?

10 I got this problem from reading Brian Hayes' article "How to Count" (Hayes, 2001). The transition matrix for a four-state (two-bit) binary counter is

	00	01	10	11
00	0	1	0	0
01	0	0	1	0
10	0	0	0	1
11	1	0	0	0

The matrix shows by rows:

- If the current count is 00, then adding 1 gives a count of 01.
- If the current count is 01, then adding 1 gives a count of 10.
- If the current count is 10, then adding 1 gives a count of 11.
- If the current count is 10, then adding 1 gives a count of 00.

Consider the matrix as an adjacency matrix for a relation $\vdash$ between the 4 counters ab, a, $b \in \mathbb{B}$. Is the $\vdash$ relation between states reflexive? symmetric? antisymmetric? transitive? Explain your answers.

11 I play Scrabble and Words with Friends. Players have 7 tiles in their rack. Each tile contains a letter worth a certain number of points or a 0 point blank.

11.1 Pretend all 7 tiles hold different letters. How many different permutations are there?

11.2 Pretend two of the letters are identical and the other 5 are all different. How many different permutations are there?

11.3 As you play the game, the normal alphabetical order

$$\langle A, B, C, D, E, F, G \rangle$$

is permuted as you search for words. Use cycle notation to write the permutations below of the normal order. How many cycles are in each permutation?

11.3.1 $\langle A, B, C, D, E, F, G\rangle$

11.3.2 $\langle G, F, E, D, C, B, A\rangle$

11.3.3 $\langle D, C, B, A, G, F, E\rangle$

11.4 Given the permutations below, written in cycle notation, list the letters in the order they appear in the permutation.

11.4.1 $[A, B, C, D, E, F, G]$

11.4.2 $[A, G, B]\,[D, F, C]\,[E]$

11.4.3 $[D, C]\,[B, A, G]\,[F, E]$

11.5 The table below shows the distribution of the 100 tiles in Scrabble.

	Tiles and their occurrences							
Points	1	2	3	4	6	8	9	12
0		␣						
1				L, S, U	N, R, T	O	A, I	E
2			G	D				
3		B, C, M, P						
4		F, H, V, W, Y						
5	K							
8	J, X							
10	Q, Z							

11.6 Pretend all of the tiles are different, for example, there are two different B's: B_1 and B_2. In how many can you draw 7 tiles?

11.7 In how many ways can you draw $ALGEBRA$? Rule out the use of a blank ␣.

Consider the in-order (left, root, right) tree traversal below. What permutation converts it to the pre-order (root, left, right) traversal?

```
      7
    /   \
  5       13
 / \
2   3
```

$\xrightarrow{\text{in-order}}$ $[2, 5, 3, 7, 13]$

```
      7
    /   \
  5       13
 / \
2   3
```

$\xrightarrow{\text{pre-order}}$ $[7, 5, 2, 3, 13]$

14. Equivalences: How to partition a set

If a body gives off the energy L in the form of radiation, its mass diminishes by L/c^2.

Albert Einstein, in the paper "Does the inertia of a body depend upon its energy-content?" September 27, 1905

Russell is referring to the transitive, symmetric, and reflexive properties of a relation. If a relation possesses all of these, it is an *equivalence*.

Relations which possess these [equivalence] properties are an important kind, and it is worthwhile to note that similarity is one of this kind of relations.

Bertrand Russell, Principles of Mathematics, 1903

Testing input/output behavior

Pretend that you've managed to write and compile a program that you think will run on your computer. You need to test that it computes correct results. Although you've been testing your work throughout development, you must now show that it passes all test cases. A *black-box* test focuses on the input/output behavior of software. Such a test starts with a known behavior and determines if the software duplicates it.

The testing example is borrowed from (Kaner et al., 1999). How many possible tests could be run?

A first-level black-box test partitions the input space into valid and invalid values. For instance, suppose valid input to your program P are a pair of integers (a, b) chosen from -99 to 99. You can partition the integers into two sets:

1. Valid input: $a = -99, -98, \ldots, 98, 99$ and $b = -99, -98, \ldots, 98, 99$
2. Invalid input: $|a| > 99$ or $|b| > 99$ (ignoring non-integral input)

In software testing it is useful to identify boundaries between valid and invalid input. You can refine the partition given above into:

1. Valid input:
   (a) Boundaries: $n = -99, 99$
   (b) Interior values $n = -98, \ldots, 98$
2. Invalid input:
   (a) Boundaries: $n = -100, 100$

(b) Exterior values $n < -100$ or $n > 100$

This partitioning of input defines an equivalence relation on the integers. Output partition testing may be possible too. But these are topics best learned in a study of software testing. Some basic equivalence relations are described below.

Arithmetic expressions

Have you every had a computer graded exam mark your answer wrong because you typed in 1/2 instead of 0.5? Worst yet, did your third grade teacher mark you down because you wrote $1/\sqrt{2}$ instead of $\sqrt{2}/2$? The equality of some arithmetic expressions are easy to prove. Others are harder, for instance, a formula involving π is

$$\frac{\pi^2}{6} = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \cdots = \sum_{k=1}^{\infty} \frac{1}{k^2}$$

In any event, whether easy or hard to prove, equivalence of arithmetic expressions partition all arithmetic expressions into classes where each expression in a class is equivalent to all other expressions in the class.

Boolean expressions

There are several equivalent Boolean expressions for `True`.

$$\texttt{True} \equiv \mathrm{P} \vee \neg \mathrm{P} \equiv (\mathrm{P} \wedge (\mathrm{P} \Rightarrow \mathrm{Q})) \Rightarrow \mathrm{Q}$$

Expressions that are always `True`, such as *modus ponens*, are used in proofs. Another useful equivalence is the connection between *conditional* and *disjunction* statements.

$$(\mathrm{P} \Rightarrow \mathrm{Q}) \equiv (\neg \mathrm{P} \vee \mathrm{Q})$$

In any event, this idea of equivalence between Boolean expressions partitions all Boolean expressions into classes where each expression in a class is equivalence to all others in the class.

Congruence modulo m

$\mathbb{Z}_m$ is the set *integers mod m* or *modular integers*. Each modular integer is an equivalence class ordinary integers.

$$\mathbb{Z}_m = \{0,\ 1,\ 2,\ 3, \ldots,\ (m-1)\}$$

The modular number $k \pmod{m}$ represents the set of numbers that have reminder k when divided by m.

$$k \pmod{m} = \{k,\ k \pm m,\ k \pm 2m,\ k \pm 3m,\ k \pm 4m, \ldots\}$$

For instance,

$$3 \pmod{5} = \{3,\ 3 \pm 5,\ 3 \pm 10,\ 3 \pm 15,\ 3 \pm 20, \ldots\}$$

Congruence mod 2 partitions the integers into the "evens" and "odds."

$0 = \{0,\ \pm 2,\ \pm 4,\ \pm 6,\ \pm 8,\ \pm 10, \ldots\}$

$1 = \{1,\ 1 \pm 2,\ 1 \pm 4,\ 1 \pm 6,\ 1 \pm 8,\ 1 \pm 10, \ldots\}$

Some people use syntatic sugar such as $[0]_2$ and $[1]_2$ to denote the sets of even and odd integers.

Modulo 3 partitions the integers into three sets determined by their remainders when divided by 3.

$0 = \{0,\ \pm 3,\ \pm 6,\ \pm 9,\ \pm 12,\ \pm 15, \ldots,\ 3n, \ldots\}$

$1 = \{1,\ 1 \pm 3,\ 1 \pm 6,\ 1 \pm 9,\ 1 \pm 12,\ 1 \pm 15, \ldots,\ 1 \pm 3n, \ldots\}$

$2 = \{2,\ 2 \pm 3,\ 2 \pm 6,\ 2 \pm 9,\ 2 \pm 12,\ 2 \pm 15, \ldots,\ 2 \pm 3n, \ldots\}$

Definition 30: Congruence mod m

Let a and b be integers ($a, b \in \mathbb{Z}$) and let m be a natural number. Then

$$a \equiv b \pmod{m} \quad \text{if and only if} \quad a - b \quad \text{is a multiple of } m.$$

To say "$a - b$ is a multiple of m" is the same as:

- There is an integer c such that $(a - b) = cm$.
- m divides $a - b$, in mathematical notation $a \mid (a - b)$.
- The remainders of a and b, when divided by m, are equal, that is $\bmod m = b \bmod m$.

Fractions & Projective spaces

In third grade you started to learn about fractions $\frac{a}{b}$, provided $b \neq 0$. You were taught that a fractional value could be represented in many different ways. For instance,

$$\frac{2}{3} = \frac{4}{6} = \frac{6}{9} = \frac{8}{12} = \frac{10}{15}$$

It is useful to consider a fraction, a/b to be an ordered pair (a, b). Equality between fractions can then be thought of as equality between points. For instance,

$$(2, 3) = (4, 6) = (6, 9) = (8, 12) = (10, 15)$$

Notice these points lie on the line $3x = 2y$ with slope $m = 3/2$ and y-intercept $b = 0$.

Equality is an equivalence relation. Equality of fractions is tested by cross-multiplication. That is,

$$\frac{a}{b} = \frac{c}{d} \quad \text{if } ad = cb.$$

A neat idea is to treat $\frac{a}{b}$ as a point (a, b) in two-dimensional *projective space*. The idea behind *projective space* is that $(a, b) \equiv (\lambda a, \lambda b)$ for all non-zero values of λ.

$$(\forall\, \lambda \neq 0)((a, b) \equiv (\lambda a, \lambda b))$$

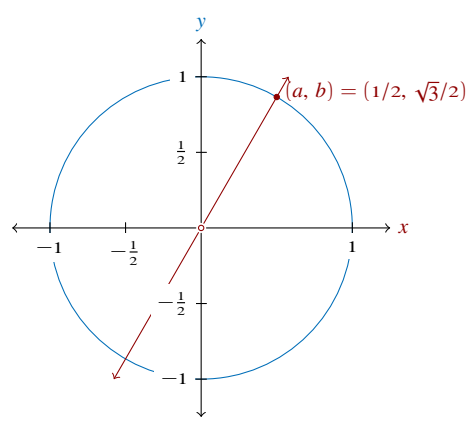
Projections of three-dimensional space onto two-dimensions are how pictures are made; they are an important step in the computer graphics pipeline.

That is, all points (*except* the origin $(0, 0)$) along the line defined by the origin $(0, 0)$ and (a, b) are equivalent. However, the study of projective geometry is beyond the scope of these notes.

Definition 31: Homogeneity of points

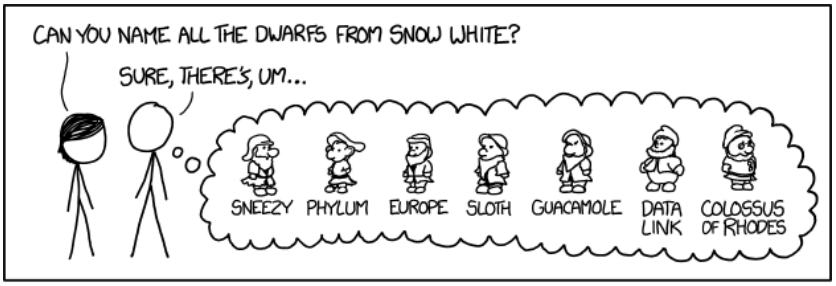
Let (a, b) and (c, d) be two ordered pairs of integers, neither of which is the origin $(0, 0)$. Then,

$$(a, b) \propto (c, d) \quad \text{if } ad = bc.$$



All points on the punctured line through $(0, 0)$ and (a, b) are equivalent. Think of them being projected onto the unit circle.

Cardinality & equinumerosity



I HAVE THIS PROBLEM WHERE ALL SETS OF SEVEN THINGS ARE INDISTINGUISHABLE TO ME.

Seven

Cardinality establishes an equivalence on the collection of all sets. For instance, when sets have the same cardinality they can be considered equivalent. The set of integers mod 5 can be considered indistinguishable from some of the SpongeBob SquarePants cast.

$$\mathbb{Z}_5 = \{0,\ 1,\ 2,\ 3,\ 4\} \equiv \{$$  $$\} = \mathbb{SBSP}$$

By cardinality, the natural numbers, the integers, and the rational numbers are indistinguishable.

$$\mathbb{N} \equiv \mathbb{Z} \equiv \mathbb{Q} \quad \text{because } \aleph_0 = |\mathbb{N}| = |\mathbb{Z}| = |\mathbb{Q}|$$

Pass a Quiz: Basics on equivalences

Take a quiz on page 330 to check your understanding. You can return to here from the quiz.

Definition of equivalence

Definition 32: Equivalence relation

An equivalence $\equiv$ *satisfies three conditions:*

1. *Reflexive law: For all a, $a \equiv a$.*

$$(\forall\ a)(a \equiv a)$$

2. *Symmetric law: For all a and b, if $a \equiv b$, then $b \equiv a$.*

$$(\forall\ a,\ b,\ )(a \equiv b) \Rightarrow (b \equiv a)$$

3. *Transitive law: For all a, b and c, if $a \equiv b$ and $b \equiv c$, then $a \equiv c$.*

$$(\forall\ a,\ b,\ c)(((a \equiv b) \wedge (b \equiv c)) \Rightarrow (a \equiv c))$$

Like the warning in the induction chapter, to *pass the class* there are two basic relations you must be able to demonstrate are equivalences.

1. Congruence mod m is an equivalence: It is reflexive, symmetric, and transitive.

2. Homogeneity of points is an equivalence: It is reflexive, symmetric, and transitive.

Theorem 37: Congruence mod m is an equivalence

Let a, b be integers, and let $m \in \mathbb{N}$ be a natural number. Then

$$a \equiv b \pmod{m} \quad \text{is an equivalence relation.}$$

Proof: Congruence mod m is an equivalence

You must show congruence mod m is reflexive, symmetric, and transitive.

- *Congruence is reflexive. $a - a$ is a multiple of m.*

$$a - a = 0 \cdot m \Rightarrow a \equiv a \pmod{m}$$

- *Congruence is symmetric. If $a - b$ is a multiple of m, then $b - a$ is a multiple of m.*

$$a - b = c \cdot m \Rightarrow b - a = -c \cdot m$$

That is,

$$a \equiv b \pmod{m} \Rightarrow b \equiv a \pmod{m}$$

- *Congruence is transitive. If both $a - b$ and $b - c$ are a multiples of m, then $(a - b) + (b - c) = a - c$ is a multiple of m. That is,*

$$(a \equiv b \pmod{m} \wedge b \equiv c \pmod{m}) \Rightarrow a \equiv c \pmod{m}$$

Congruence mod m partitions the integers into m equivalence classes. For instance, congruence mod 5 partitions the integers into 5 equivalence classes. equivalence classes

$$\begin{aligned}
0 &= \{0,\ \pm 5,\ \pm 10,\ \pm 15,\ \pm 20, \ldots,\ \pm 5n, \ldots\} \\
1 &= \{1 \pm 5,\ 1 \pm 10,\ 1 \pm 15,\ 1 \pm 20, \ldots,\ 1 \pm 5n, \ldots\} \\
2 &= \{2 \pm 5,\ 2 \pm 10,\ 2 \pm 15,\ 2 \pm 20, \ldots,\ 2 \pm 5n, \ldots\} \\
3 &= \{3 \pm 5,\ 3 \pm 10,\ 3 \pm 15,\ 3 \pm 20, \ldots,\ 3 \pm 5n, \ldots\} \\
4 &= \{4 \pm 5,\ 4 \pm 10,\ 4 \pm 15,\ 4 \pm 20, \ldots,\ 4 \pm 5n, \ldots\}
\end{aligned}$$

The adjacency matrix for congruence mod 3 on the digits is shown below. A permutation has been applied to the natural order to show blocks of equiva-

lent digits.

Congruence mod 3 on $\mathbb{D}$

	0	3	6	9	1	4	7	2	5	8
0	1	1	1	1	0	0	0	0	0	0
3	1	1	1	1	0	0	0	0	0	0
6	1	1	1	1	0	0	0	0	0	0
9	1	1	1	1	0	0	0	0	0	0
1	0	0	0	0	1	1	1	0	0	0
4	0	0	0	0	1	1	1	0	0	0
7	0	0	0	0	1	1	1	0	0	0
2	0	0	0	0	0	0	0	1	1	1
5	0	0	0	0	0	0	0	1	1	1
8	0	0	0	0	0	0	0	1	1	1

Theorem 38: Homogeneity of points

Let (a, b) and (c, d) ordered pairs of integers. Then

$$(a, b) \propto (c, d) \quad \text{is an equivalence}$$

where $(a, b) \propto (c, d)$ means $ad = cb$.

Proof: Homogeneity of points

You must show homogeneity is reflexive, symmetric, and transitive.

- *Homogeneity is reflexive.* $(a, b) \propto (a, b)$. *That is,*

$$ab = ba$$

- *Homogeneity is symmetric. If* $(a, b) \propto (c, d)$, *then* $(c, d) \propto (a, b)$. *That is,*

$$(ad = bc) \Rightarrow (cb = da)$$

- *Homogeneity is transitive. If both* $(a, b) \propto (c, d)$ *and* $(c, d) \propto (e, f)$, *then* $(a, b) \propto (e, f)$. *That is, suppose*

$$(ad = bc) \wedge (cf = ed)$$

Assume $f \neq 0$ *and multiply both sides of* $ad = bc$ *by* f *to get*

$$adf = bcf = b(ed) \quad \text{or} \quad af = eb, \quad \text{provided } d \neq 0.$$

What if $d = 0$ *when* $f \neq 0$? *In this case,* $0 = ed = cf$ *at least one of* c *or* f *must be zero. By assumption, it must be* $c = 0$. *But,* c *cannot be* 0 *either because* $(c, d) = (0, 0)$ *is not a point in projective space. Therefore* $d = 0$ *and* $f \neq 0$ *leads to a contradiction.*

What if $f = 0$? *Since* $0 = cf = ed$ *it must be that* $d = 0$. *But then* $0 = ad = cb$ *and it must be that* $b = 0$. *Therefore,*

$$af = eb \quad \text{is } \texttt{True}. \text{ that is } (a, b) \propto (e, f).$$

This case: $d = 0$ and $f \neq 0$ cannot happen. The proof at the right is an example of contradiction or *reductio ad absurnum*. The assumption $d = 0$ and $f \neq 0$ leads to a contraction.

Recall, two-dimensional projective space is punctured at $(0, 0)$: The origin is not a homogeneous point: $(c, d) \neq (0, 0)$.

Partitions of a set

A set can be partitioned into disjoint subsets.

Definition 33: Disjoint sets

Sets $\mathbb{X}$ and $\mathbb{Y}$ are disjoint if they have an empty intersection.

$$\mathbb{X} \cap \mathbb{Y} = \emptyset$$

Definition 34: Partition of a set

Let $\mathbb{A} = \{\mathbb{X}_k : k = 0, 1, \ldots, (n-1)\}$ be a collection of subsets of $\mathbb{U}$. The collection is a partition of $\mathbb{U}$ if two properties hold:

1. *The sets are pairwise disjoint*

$$\mathbb{X}_i \cap \mathbb{X}_j = \emptyset \quad \text{whenever } i \neq j.$$

2. *The sets cover $\mathbb{U}$*

$$\bigcup_{k=0}^{n-1} \mathbb{X}_k = \mathbb{U}$$

Theorem 39: Partitions and equivalences are equivalent

If $\equiv$ is an equivalence on $\mathbb{U}$, then the equivalence classes of $\equiv$ partition $\mathbb{U}$.
If $\mathbb{A}$ is a partition of $\mathbb{U}$, then $\equiv$ is an equivalence relation, where $a \equiv b$ if and only if both a and b are members of the same subset of the partition.

Simple proof, need to write it

Proof: Partitions and equivalences are equivalent

TDB

For small cardinality sets it is possible to list all partitions.

Partitions of a set into subsets

Set	Partitions with 1 subsets	2 subsets	3 subsets	4 subsets
$\{0\}$	$\{\{0\}\}$			
$\{0, 1\}$	$\{\{0, 1\}\}$	$\{\{0\}, \{1\}\}$		
$\{0, 1, 2\}$	$\{\{0, 1, 2\}\}$	$\{\{0\}, \{1, 2\}\}$ $\{\{1\}, \{0, 2\}\}$ $\{\{2\}, \{0, 1\}\}$	$\{\{0\}, \{1\}, \{2\}\}$	
$\{0, 1, 2, 3\}$	$\{\{0, 1, 2, 3\}\}$	$\{\{0, 1, 2\}, \{3\}\}$ $\{\{0, 3\}, \{1, 2\}\}$ $\{\{0\}, \{1, 2, 3\}\}$ $\{\{1, 3\}, \{0, 2\}\}$ $\{\{1\}, \{0, 2, 3\}\}$ $\{\{2, 3\}, \{0, 1\}\}$ $\{\{2\}, \{0, 1, 3\}\}$	$\{\{0\}, \{1, 2\}, \{3\}\}$ $\{\{1\}, \{0, 2\}, \{3\}\}$ $\{\{2\}, \{0, 1\}, \{3\}\}$ $\{\{0\}, \{1\}, \{2, 3\}\}$ $\{\{0\}, \{1, 3\}, \{2\}\}$ $\{\{0, 3\}, \{1\}, \{2\}\}$	$\{\{0\}, \{1\}, \{2\}, \{3\}\}$

The number ways to partition an n-element set into k subsets is a Stirling's numbers of the second kind, named by $\left\{ {n \atop k} \right\}$. Stirling's numbers of the second kind can be computed by the recurrence

$$\left\{ {n \atop k} \right\} = k \left\{ {n-1 \atop k} \right\} + \left\{ {n-1 \atop k-1} \right\}$$

Check the following arithmetic to verify the values in table to the right.

$$\left\{ {4 \atop 3} \right\} = 3 \left\{ {3 \atop 3} \right\} + \left\{ {3 \atop 2} \right\} = 3 \cdot 1 + 3$$
$$\left\{ {5 \atop 3} \right\} = 3 \left\{ {4 \atop 3} \right\} + \left\{ {4 \atop 2} \right\} = 3 \cdot 6 + 7$$
$$\left\{ {7 \atop 5} \right\} = 5 \left\{ {6 \atop 5} \right\} + \left\{ {6 \atop 4} \right\} = 5 \cdot 15 + 65$$

with boundary conditions

$$\left\{ {n \atop 0} \right\} = 0 \text{ for } n > 0 \quad \text{and} \quad \left\{ {n \atop n} \right\} = 1, \text{ for } n \geq 0$$

The notation $\left\{ {n \atop m} \right\}$ is said "n subset m." The first few Stirling's numbers of the second kind are shown in the table below.

Stirling's second rectangle

Stirling Numbers of the Second Kind $\left\{ {n \atop k} \right\}$

		Subset k									
		0	1	2	3	4	5	6	7	8	9
	0	1	0	0	0	0	0	0	0	0	0
	1	0	1	0	0	0	0	0	0	0	0
	2	0	1	1	0	0	0	0	0	0	0
	3	0	1	3	1	0	0	0	0	0	0
n	4	0	1	7	6	1	0	0	0	0	0
	5	0	1	15	25	10	1	0	0	0	0
	6	0	1	31	90	65	15	1	0	0	0
	7	0	1	63	301	350	140	21	1	0	0
	8	0	1	127	966	1701	1050	266	28	1	0
	9	0	1	255	3025	7770	6951	2646	426	36	1

Here are some color-coded examples of partitions. Let $\mathbb{X} = \{0, 1, 2\}$ so that $n = |\mathbb{X}| = 3$. In this case there are 5 different equivalence relations (partitions) on $\mathbb{X}$.

	Partitions with	
1-subset	2-subsets	3-subsets
$\{\{0, 1, 2\}\}$	$\{\{0\}, \{1, 2\}\}$	$\{\{0\}, \{1\}, \{2\}\}$
	$\{\{1\}, \{0, 2\}\}$	
	$\{\{2\}, \{0, 1\}\}$	

When $\mathbb{X}' = \{0, 1, 2, 3\}$ and $n = 4$ there are 15 different equivalence relations (partitions) that can be defined on $\mathbb{X}'$.

1-subset	Partitions with 2-subsets	3-subsets	4-subsets
$\{\{0, 1, 2, 3\}\}$	$\{\{3\}, \{0, 1, 2\}\}$	$\{\{0\}, \{1, 2\}, \{3\}\}$	$\{\{0\}, \{1\}, \{2\}, \{3\}\}$
	$\{\{0\}, \{1, 2, 3\}\}$	$\{\{1\}, \{0, 2\}, \{3\}\}$	
	$\{\{1\}, \{0, 2, 3\}\}$	$\{\{2\}, \{0, 1\}, \{3\}\}$	
	$\{\{2\}, \{0, 1, 3\}\}$	$\{\{0\}, \{1\}, \{2, 3\}\}$	
	$\{\{0, 3\}, \{1, 2\}\}$	$\{\{0\}, \{1, 3\}, \{2\}\}$	
	$\{\{1, 3\}, \{0, 2\}\}$	$\{\{0, 3\}, \{1\}, \{2\}\}$	
	$\{\{2, 3\}, \{0, 1\}\}$		

If you study the partitions above , you'll discover how to compute $\left\{{4 \atop k}\right\}$ from $\left\{{3 \atop k}\right\}$ and $\left\{{3 \atop k-1}\right\}$.

Consider the seven $k = 2$-subset partitions of $\{0, 1, 2, 3\}$. They were constructed from the 1-subset and 2-subset partitions of $\{0, 1, 2\}$.

- The union of $\{\{3\}\}$ with the 1-subset partition $\{\{0, 1, 2\}\}$ of $\{0, 1, 2\}$ forms a 2-subset partition of $\{0, 1, 2, 3\}$.

$$\{\{0, 1, 2\}\} \to \{\{0, 1, 2\}\} \cup \{\{3\}\} = \{\{0, 1, 2\}, \{3\}\}$$

- The union of $\{3\}$ with each set in a 2-subset partition of $\{0, 1, 2\}$ forms six more 2-subset partitions of $\{0, 1, 2, 3\}$.

$$\begin{aligned}
\{\{0\}, \{1, 2\}\} &\to \{\{0, 3\}\} \cup \{\{1, 2\}\} \\
&\to \{\{0\}\} \cup \{\{1, 2, 3\}\} \\
\{\{1\}, \{0, 2\}\} &\to \{\{1, 3\}\} \cup \{\{0, 2\}\} \\
&\to \{\{1\}\} \cup \{\{0, 2, 3\}\} \\
\{\{2\}, \{0, 1\}\} &\to \{\{2, 3\}\} \cup \{\{0, 1\}\} \\
&\to \{\{2\}\} \cup \{\{0, 1, 3\}\}
\end{aligned}$$

This construction is general. It leads to the Stirling's rule of the second kind.

Theorem 40: Stirling's rule of the second kind

Let $\left\{ {n \atop k} \right\}$ be the number of partitions of an n-element set into k-subsets. Then,

$$\left\{ {n \atop k} \right\} = k\left\{ {n-1 \atop k} \right\} + \left\{ {n-1 \atop k-1} \right\}$$

The partition $\mathbb{A}$ is a collection of disjoint subsets that cover $\mathbb{X}$.

Proof: Stirling's rule of the second kind

Let $\mathbb{X} = \{x_0, x_1, \ldots, x_{n-1}\}$ and $\mathbb{X}' = \{x_0, x_1, \ldots, x_{n-1}, x_n\}$ be sets containing n and $(n+1)$ elements. There are two ways to construct a k-subset partition of $\mathbb{X}'$.

1. *Every partition $\mathbb{A}$ of $\mathbb{X}$ into $(k-1)$ subsets generates a partition of $\mathbb{X}'$ into k subsets. The construction is: Include (union) the singleton $\{\{x_n\}\}$ with the partition $\mathbb{A}$ creating a k subset partition of $\mathbb{X}'$. The function that computes the number of $(k-1)$ subset partitions of $\mathbb{X}$ is named $\left\{ {n-1 \atop k-1} \right\}$.*
2. *Every partition $\mathbb{A}$ of $\mathbb{X}$ into k subsets generates k partitions of $\mathbb{X}'$ into k subsets. The construction is: Include (union) the singleton $\{x_n\}$ with each of the k subsets in $\mathbb{A}$. That is, each of the $\left\{ {n-1 \atop k} \right\}$ partitions of $\mathbb{X}$ generates k partitions of $\mathbb{X}'$ into k subsets.*

Therefore,

$$\left\{ {n \atop k} \right\} = k\left\{ {n-1 \atop k} \right\} + \left\{ {n-1 \atop k-1} \right\}$$

Homework Questions

Use your time outside of class to solve these problems.

1 Consider the $16 = 2^{2\times 2}$ relations on the bits $\mathbb{B} = \{0, 1\}$. These are listed on page 181.

Which of them are equivalence relations?

2 What does $a \equiv b \pmod 1$ mean?

3 What does $a \equiv b \pmod 0$ mean?

4 Consider two types of natural numbers: Odious and evil numbers. Odious numbers have an odd number of 1's in their binary expansion. Evil numbers have an even number of 1's in their binary expansion. Show that Odious and evil numbers partition the natural numbers.

5 Show the relations below are equivalences

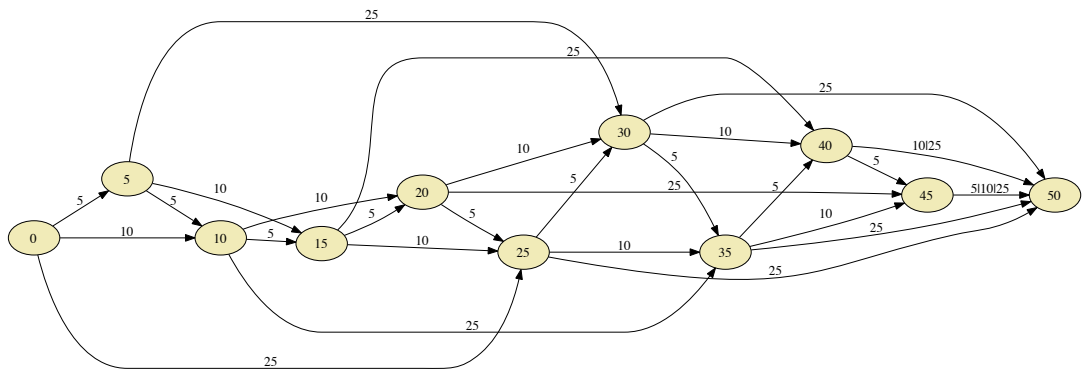
5.1 Congruence mod m: $a \equiv b \pmod m$ for $a, b \in \mathbb{Z}$ and $m \in \mathbb{N}$, $m > 0$.

5.2 Homogeneity of points: $(a, b) \propto (c, d)$, for $a, b, c, d \in \mathbb{Z}$.

5.3 Parallel lines: $\ell_1 \| \ell_2$ for planar lines.

6 There are five equivalence relations on $\mathbb{X} = \{a, b, c\}$. What are they?

7 A candy machine takes nickels, dimes and quarters. When 50¢ is deposited a candy bar is dispensed along with change if necessary. The diagram below shows the transitions from state-to-state as money is put in the machine.



Say that two paths p_0 and p_1 in the transition graph are path equivalent they start in the same state s and end in the same state t. For instance,

$$p_0 : 5 \xrightarrow{25} 30 \xrightarrow{10} 40$$

is equivalent to

$$p_1 : 5 \xrightarrow{10} 15 \xrightarrow{10} 25 \xrightarrow{10} 35 \xrightarrow{5} 40$$

Prove path equivalence is an equivalence relation.

8 Let $f : \mathbb{X} \to \mathbb{Y}$ be a function. Let $a, b \in \mathbb{X}$, and write $a \equiv b$ if $f(a) = f(b)$. Show that $\equiv$ is an equivalence relation.

9 Let m and a be integers. Write $m \perp a$ if 1 is the only integer that divides both m and a. If $m \perp a$, say m and a are relatively prime. Is this relation reflexive, symmetric, antisymmetric, or transitive?

10 Consider the set of all people in the world whose first name is written using the English alphabet. Pretend you group these people based on the first letter of their first name: All people whose name starts "A..." are placed in one set; All people whose name starts "B..." are placed in another set, and so on. (Notice these are sets of people, not sets of names.) Does this *partition* the group of people? Is this an equivalence relation on the people? Explain your answer.

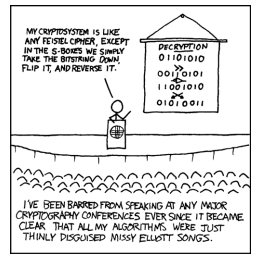
Pass a Quiz: Summative exam # 5 on relations: orders and equivalences

Take a quiz on page 331 to check your understanding. You can return to here from the quiz.

15. Modular arithmetic: Randomness and cryptology

Random numbers should not be generated with a method chosen at random.

Donald Knuth



Cryptography

Any one who considers arithmetical methods of producing random digits is, of course, in a state of sin.

John von Neumann, "Various techniques used in connection with random digits" by John von Neumann in Monte Carlo Method (1951), ed. A.S. Householder, G.E. Forsythe, and H.H. Germon

It is often useful to generate a sequence $\vec{R}$ of random numbers. Random numbers are used in cryptography, computer simulations of phenomena, evaluation of integrals, sampling a population, and many other applications.

Random Numbers

Modular recurrences are a common way to generate pseudo-random numbers. Let

$$\vec{R} = \langle r_0, r_1, r_2, r_3, \ldots \rangle$$

be a sequence of numbers generated by a recurrence equation of the form

$$r_n = (ar_{n-1} + b) \bmod m.$$

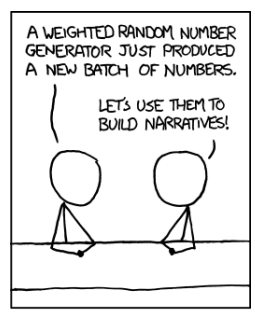
The value r_0 is called the *seed* or *initial condition*. The modulus m and parameters a and b are (usually) non-zero natural numbers. A "random" sequence of this type will repeat after m or fewer iterations. For instance, consider the recurrence

$$r_n = (3r_{n-1} + 1) \bmod 5 \quad \text{with } r_0 = 0.$$

Convince yourself this recurrence generates the sequence

$$\vec{R} = \langle 0, 1, 4, 3, 0, \ldots \rangle$$

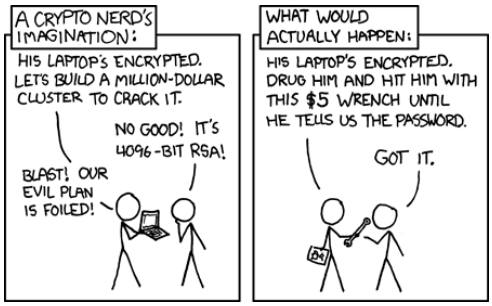
which has *period* 4.



Sports

This recurrence is a linear congruential generator model of randomness.

Cryptology



Security

Shift right or left it hardly matters.

Caesar ciphers are constructed by shifting each letter k places to the right, for some parameter k. For example, the Caesar cipher $k = 5$ on the uppercase English alphabet $\mathbb{E} = \{A, B, C, \ldots, X, Y, Z\}$ is

0	1	2	3	4	5	6	7	8	9	10	11	12	numerical index
A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*	clear text
F	*G*	*H*	*I*	*J*	*K*	*L*	*M*	*N*	*O*	*P*	*Q*	*R*	coded text

13	14	15	16	17	18	19	20	21	22	23	24	25	numerical index
N	*O*	*P*	*Q*	*R*	*S*	*T*	*U*	*V*	*W*	*X*	*Y*	*Z*	clear text
S	*T*	*U*	*V*	*W*	*X*	*Y*	*Z*	*A*	*B*	*C*	*D*	*E*	coded text

Given an index n pointing to a letter x and a shift k, encode x by using the letter at position $n + k \bmod 26$. . The encryption function is

$$e(n, k) = (n + k) \bmod 26$$

And to decrypt a letter compute

$$d(n, k) = (n - k) \bmod 26$$

For instance, the word "CAFE" can be written as the number "2054". The encoding of "CAFE" is "HFKJ"

$$\begin{aligned} e(2, 5) &= (2 + 5) \bmod 26 = 7 &&= H \\ e(0, 5) &= (0 + 5) \bmod 26 = 5 &&= F \\ e(5, 5) &= (5 + 5) \bmod 26 = 10 &&= K \\ e(4, 5) &= (4 + 5) \bmod 26 = 9 &&= J \end{aligned}$$

A Caesar cipher is easy to break. A brute force approach simply searches 25 possible shifts to decode a message.

The HASKELL code below, from Rosetta Code, implements a Caesar cipher.

```
import Data.Char (ord, chr)
import Data.Ix (inRange)

caesar :: Int -> String -> String
caesar k = map f
  where
    f c
      | inRange ('a','z') c = tr 'a' k c
      | inRange ('A','Z') c = tr 'A' k c
      | otherwise = c

unCaesar :: Int -> String -> String
```

```
unCaesar k = caesar (-k)

-- char addition
tr :: Char -> Int -> Char -> Char
tr base offset char = chr ( ord base + (ord char - ord base + offset)) `mod` 26
```

In a difference context, the real numbers mod π appear in the study of *trigonometry*. For instance, the $\cos(x)$ and $\sin(x)$ functions are *periodic* mod 2π.

$$\cos(x) = \cos(x \pm 2\pi n) \quad \text{and} \quad \sin(x) = \sin(x \pm 2\pi n)$$

This one dimensional map is the most simple affine transformation: Affine maps can be usefully extended to higher dimensional spaces.

Affine Ciphers

An affine map is a function of the form

$$y = ax + b.$$

An affine cipher on the uppercase English alphabet encodes x as $y = (ax + b) \bmod m$. That is, letting $m = 26$, the encoding function is

$$y = e(x, a, b) = (ax + b) \bmod 26$$

for some parameters a and b.

To decode a message, you must solve the equation below for x

$$y = (ax + b) \bmod 26$$

Remember, it's modular arithmetic here. All numbers are (modular) integers, including a^{-1}.

If m is a prime number, then it can be shown that every integer a from 1 to $m - 1$ has an inverse $a^{-1} \bmod m$. If m is not a prime number, then there is a unique solution to $ax = y - b \bmod m$ if and only if $\gcd(a, m) = 1$.

From basic algebra you know how to find x: Subtract b from both sides. Then multiply both sides by the reciprocal of a, which in basic algebra is defined by the equation

$$\frac{1}{a} \cdot a = 1$$

Also, you know the reciprocal can be written exponentially, which is better notation here.

$$\frac{1}{a} = a^{-1} \quad \text{for } a \neq 0.$$

To solve the equation

$$y = ax + b$$

you must compute

$$x = a^{-1}(y - b) \bmod 26$$

One problem is: Not every value of $a \neq 0$ will have a reciprocal (a multiplicative inverse) $a^{-1} \bmod 26$.

For instance, 13 does not have an inverse mod 26 The equation $13x = (y - b) \bmod 26$ only has a solution when $y - b = 0$ or 13 mod 26. In these cases, there are many solutions.

$$13x = 0 \bmod 26 \quad \text{is } \texttt{True} \text{ for } x = 0, 2, 4, \ldots, 24.$$

And,

$$13x = 13 \bmod 26 \quad \text{is } \texttt{True} \text{ for } x = 1, 3, 5, \ldots, 25.$$

On the other hand, $3x = (y - b) \bmod 26$ has a unique solution for each value of $y - b$ from 0 to 25.

You can verify that $x = 9(y - b) \bmod 26$ by substitution:

$$\begin{aligned} 3x &= 3 \cdot 9(y - b) \\ &= 27(y - b) \\ &= (26 + 1)(y - b) \bmod 26 \\ &= (y - b) \bmod 26 \end{aligned}$$

Modular arithmetic does work this way, but we should understand that it does by proving that it does.

Pass a Quiz: Basic cryptography concepts

Take a quiz on page 333 to check your understanding. You can return to here from the quiz.

Modular arithmetic

Recall from the notes on arithmetic, given a positive integers m, the modular integers are values in the set

$$\mathbb{Z}_m = \{0,\ 1,\ 2,\ 3, \ldots,\ (m-1)\}$$

Arithmetic on modular integers is similar to, but different from, the arithmetic you learned in third grade. When the modulus $m = p$ is a prime number, the modular integers $\mathbb{Z}_p = \{0,\ 1,\ 2,\ 3, \ldots,\ (p-1)\}$ form a field: Addition, subtraction, multiplication and division all well-defined and closed operations that obey the traditional arithmetic laws outlined in the notes on numerical calculations on page 27. Let's start with addition.

Modular addition

Suppose you work for 8 hours starting at 9 o'clock. Your work ends at $9 + 8 = 17$ o'clock, which may be a fine answer in cultures were a 24 hour clock is use. But, today, in America, we would say, work was over at $17 - 12 = 5$ o'clock.

Definition 35: Addition modulo m

Let $a,\ b \in \mathbb{Z}_m$. Addition modulo m is defined by

$$(a+b) \bmod m = \begin{cases} a+b & \text{if } a+b < m \\ a+b-m & \text{if } a+b \geq m \end{cases}$$

For instance, here are examples of addition modulo 12 and 17.

$$\begin{aligned} (7+3) \bmod 12 &= 10 \\ (8+11) \bmod 12 &= 19 - 12 = 7 \\ (7+3) \bmod 17 &= 10 \\ (8+11) \bmod 17 &= 19 - 17 = 2 \end{aligned}$$

Consider sums of numbers in the set $\mathbb{Z}_m = \{0,\ 1,\ 2, \ldots,\ (m-1)\}$. The largest sum is $(m-1) + (m-1) = 2m - 2$. The definition guarantees the sum $a + b$ of any two modular integers will again be a modular integer. If $a + b < m$, then $(a+b) \bmod m = a + b \in \mathbb{Z}_m$, else if $a + b \geq m$, then

$$(a+b) \bmod m = a + b - m \leq (2m-2) - m = m - 2 \in \mathbb{Z}_m$$

Here's the mod 9 addition table. See how values of sums cycle from row-to-row and column-to-column. Also notice the 0 in the table: These

identify *additive inverse* pairs: A number and its negative.

+	0	1	2	3	4	5	6	7	8
0	0	1	2	3	4	5	6	7	8
1	1	2	3	4	5	6	7	8	0
2	2	3	4	5	6	7	8	0	1
3	3	4	5	6	7	8	0	1	2
4	4	5	6	7	8	0	1	2	3
5	5	6	7	8	0	1	2	3	4
6	6	7	8	0	1	2	3	4	5
7	7	8	0	1	2	3	4	5	6
8	8	0	1	2	3	4	5	6	7

Recall, from the quotient-remainder theorem when an integer a is divided by a (positive) integer m, a quotient q and remainder r are computed. Moreover, the remainder lies between 0 and $(m-1)$: $0 \le r \le (m-1)$. Recall, from the notes on functions $r = a \bmod m$ is the remainder when a is divided by m. Therefore, integers modulo m can be thought of as the set of possible remainders upon division by m.

For small values of a and m, you can compute r in your head. For instance,

$$15 \bmod 11 = 4 \quad \text{since } 15{=}11{+}4$$

For harder instances, you can compute r using the function

$$r = r(a, m) \to a - m\left\lfloor \frac{a}{m} \right\rfloor$$

For instance, the remainder when -30 is divided by 13 is

$$-30 \bmod 13 = -30 - 13\left\lfloor \frac{-30}{13} \right\rfloor = -30 - 13(-3) = 9$$

Notice -30 and 9 are 39 units apart: Adding $39 = 3(13)$ to -30 jumps the value to 9.

It is tempting to say $-30 = -4 \bmod 13$, and -4 is *not a wrong answer* because,

$$-30 = -2 \cdot 13 - 4 = -4 \bmod 13$$

This is akin to your pedantic third grade teacher who demanded you to write $\sqrt{2}/2$ instead of $1/\sqrt{2}$.

But, arithmetic operations on $\mathbb{Z}_m$ are *closed*: Arithmetic operations on integers mod m produce results that are back in $\mathbb{Z}_m$. Therefore

$$-30 = -3 \cdot 13 + 9 = 9 \bmod 13$$

A step from -4 to 9 is 13 units long.

Notice

$$\left(-3 < \left\lfloor \frac{-30}{13} \right\rfloor < -2\right) \to \left\lfloor \frac{-30}{13} \right\rfloor = -3$$

Theorem 41: Modular addition on the integers

Let a, b, c, d be integers. If $a = b \bmod m$ and $c = d \bmod m$, then

$$(a + c) = (b + d) \bmod m$$

Proof: Modular addition on the integers

The statements $a = b \bmod m$ and $c = d \bmod m$ mean

$$(a - b) \quad \text{and} \quad (c - d) \quad \text{are multiples of } m.$$

That is,

$$(a-b) = em \quad \text{for some integer } e.$$

And,

$$(c-d) = fm \quad \text{for some integer } f.$$

Therefore,

$$\begin{aligned}(a+c)-(b+d) &= (a-b)+(c-d)\\ &= em+fm\\ &= (e+f)m\end{aligned}$$

That is, $(a+c)-(b+d)$ *is a multiple of* m*, or*

$$(a+c) = (b+d) \bmod m$$

There's not really a subtraction operation in arithmetic: There's just addition of an *additive inverse*.

Definition 36: Additive inverse (negative) modulo m

Let $a \in \mathbb{Z}_m = \{0, 1, 2, \ldots, (m-1)\}$ *be an integer mod* m*. If* $a = 0$*, then the* additive inverse *of* a *is* a*:*

$$0+0=0$$

Otherwise, the additive inverse *of* a *is* $m-a$*:*

$$a+(m-a) = m = 0 \bmod m$$

For instance, in mod 11 or mod 17 arithmetic you can write identities like:

$$5-7 = 5+(11-7) = 9 \bmod 11 \quad \text{or} \quad -15-16 = (17-15)+(17-16) = 3 \bmod 17$$

For every modulus m, for every modular integer $a \in \mathbb{Z}_m$ there is an additive inverse.

Modular multiplication

Definition 37: Multiplication modulo m

Let $a, b \in \mathbb{Z}_m$*. Multiplication modulo* m *is defined by*

$$(a \cdot b) \bmod m = a \cdot b - km \quad \text{where } k = \left\lfloor \left(\frac{a \cdot b}{m} \right) \right\rfloor$$

For a small modulus m, you can multiply mod m by hand. The mod 9

multiplication table is given below.

×	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8
2	0	2	4	6	8	1	3	5	7
3	0	3	6	0	3	6	0	3	6
4	0	4	8	3	7	2	6	1	5
5	0	5	1	6	2	7	3	8	4
6	0	6	3	0	6	3	0	6	3
7	0	7	5	3	1	8	6	4	2
8	0	8	7	6	5	4	3	2	1

The 1 numbers in the matrix identify *multipicative inverse* pairs: A number and its reciprocal.

$$2 \cdot 5 = 10 = 1 \bmod 9, \quad 4 \cdot 7 = 28 = 1 \bmod 9, \quad 8 \cdot 8 = 64 = 1 \bmod 9$$

Using exponential notation, the mod 9 multiplicative inverse of 2 is 5, the multiplicative inverse of 4 is 7, and the multiplicative inverse of 8 is 8.

$$2^{-1} = 5,\ 5^{-1} = 2, \quad 4^{-1} = 7,\ 7^{-1} = 4, \quad 8^{-1} = 8$$

On the other hand 0, 3 and 6 have no reciprocals:

$$3x \neq 1 \quad \text{and} \quad 6x \neq 1 \quad \text{for any value of } x.$$

There's not really a division operation in arithmetic: There's just multiplication by a *multiplicative inverse*, when one exists.

Caution! The multiplicative inverse of a is *not* the fraction $\frac{1}{a}$: There are *no* fractional modular integers. *Not* every modular integer will have a multiplicative inverse: 0 has no reciprocal and other numbers can fail to have multiplicative inverses too.

Definition 38: Multiplicative inverse (reciprocal) modulo m

Let $a, b \in \mathbb{Z}_m = \{0, 1, 2, \ldots, (m-1)\}$ be integers mod m. If $a \cdot b = 1 \bmod m$, then multiplicative inverse *of a is b. That is,*

$$(a \cdot b = 1 \bmod m) \Rightarrow a^{-1} = b \quad \textit{and} \quad b^{-1} = a$$

Below, in the notes on linear congruence equations, you will find a and b are reciprocals if and only if they are relatively prime: $\gcd(a, b) = 1$ or $a \perp b$.

Theorem 42: Modular multiplication on the integers

Let a, b, c, d be integers. If $a = b \bmod m$ and $c = d \bmod m$, then then $ac = bd \bmod m$

Proof: Modular multiplication on the integers

By the assumption $(a-b) = em$ *and* $(c-d) = fm$ *for some integers* e *and* f*. Therefore,*

$$\begin{aligned} ac - bd &= ac - bc + bc - bd \\ &= c(a-b) + b(c-d) \\ &= cem + bfm \\ &= (ce + bf)m \end{aligned}$$

That is, $ac - bd$ *is a multiple of* m*, or*

$$ac = bd \bmod m$$

For instance, in mod 17 or mod 18 arithmetic you can write identities like:

$$5 \cdot 7 = 35 = 1 \bmod 17 \quad \text{or} \quad 5 \cdot 7 = 35 = 17 \bmod 18$$

If $m = p$ is a prime number, then every non-zero modular integer $a \in \mathbb{Z}_p$ has is a multipicative inverse. Otherwise, if m is not prime, some modular intergers will not have a reciprocal.

Linear congruence equations

You can use modular arithmetic to solve linear congruence equations of the form.

$$ax = b \bmod m$$

To solve this equation you need to find the reciprocal of a mod m. For instance, the mod 9 multiplication table on page 279 can be used to find reciprocals when they exisits Although enlightening, creating a multiplication table is clearly not an efficient way to compute inverses.

Unlike basic algebra where, for $a \neq 0$, $ax + b = y$ always has a solution $x = a^{-1}(y-b)$, in modular algebra, there may be:

- Exactly one solution x
- Several solutions x_k, $k = 0, \ldots, (n-1)$.
- No solutions at all

You can solve the equation

$$2x = 7 \bmod 9$$

by multiplying both sides by 5

$x = (5{\cdot}2)x = (5{\cdot}7) \bmod 9 = 35 \bmod 9 = 8 \bmod 9$

And check,

$$2 \cdot 8 = 16 = 7 \bmod 9$$

We need a modern approach for computing reciprocals, so let's go back to Euclid.

Pass a Quiz: Modular arithmetic

Take a quiz on page 333 to check your understanding. You can return to here from the quiz.

The Euclidean algorithm for computing the greatest common divisor

The Euclidean algorithm computes the greatest common divisor of two natural numbers: That is, given a and m, it computes the value of the function $\gcd(a, m)$.

For small numbers you can compute $\gcd(a, m)$ in your head or with paper & pencil. For instance, 14 divides 28 and 42, and 14 is the largest divisor of them both. Therefore,

$$14 = \gcd(42, 28)$$

One way to compute $\gcd(a, m)$ is to find the prime factorization of m and a.

Computing a prime factorization can be so hard that it is used in some cryptographic algorithm, for example the RSA scheme.

$$42 = 2 \cdot 3 \cdot 7 \quad \text{and} \quad 28 = 2^2 \cdot 7$$

You can see that $2 \cdot 7 = 14$ will divide both 42 and 28, but no larger number will.

Definition 39: Greatest common divisor

The natural number g is the greatest common divisor of a and m if two conditions hold:

- *g divides both m and a ($g \mid m$ and $g \mid a$).*
- *If d divides both m and a, then d divides g ($d \leq g$).*

Here's a basic recurrence for the greatest common divisor that leads to an efficient algorithm for computing $\gcd(a, m)$.

Theorem 43: Greatest common divisor recurrence

The greatest common divisor satisfies the recurrence

$$(a \geq m) \Rightarrow [\gcd(a, m) = \gcd(m, a \bmod m)]$$

With terminating condition

$$\gcd(a, 0) = a$$

Proof: Greatest common divisor recurrence

Let a divided by m give quotient q and remainder r.

$$a = mq + r, \quad 0 \leq r < m$$

If d divides both a and m, then d also divides

$$r = a - mq = a \bmod m$$

That is, d is a common divisor of m and $a \bmod m$. On the other hand, if d divides both m and $a \bmod m$, then d also divides

$$a = mq + r = mq + a \bmod m$$

Therefore, the set of common divisors of a and m is identical to the set of common divisors of m and $a \bmod m$, and their greatest divisors must be equal.

Assuming $m \geq a$, the HASKELL function that implements this recursion is:

This is the Euclidean algorithm, although Euclid did not use the same notation.

```
gcd :: Integral a => a -> a -> a
gcd a 0 = a
gcd a m = gcd m (a `mod` m)
```

For small numbers you can compute the greatest common divisor by paper & pencil. Here's an example computation of $\gcd(73, 17)$.

See the m and r values shift down and left (southwest) at each step.

Dividend	=	Divisor	·	Quotient	+	Remainder
a	=	m	·	q	+	r
73	=	17	·	4	+	5

The last non-zero remainder is the greatest common divisor. In this case, $\gcd(73, 17) = 1$.

Here's another example that shows $4 = \gcd(92, 52)$.

Dividend	=	Divisor	·	Quotient	+	Remainder
a	=	m	·	q	+	r
92	=	52	·	1	+	40

The last non-zero remainder is the greatest common divisor. In this case, $4 = \gcd(92, 52)$.

Here are some examples of the Euclidean algorithm with data generated randomly each time the notes are compiled.

Dividend a	Divisor m	Quotient q	Remainder r
21	21	1	0

Dividend a	Divisor m	Quotient q	Remainder r
315	193	1	122
193	122	1	71
122	71	1	51
71	51	1	20
51	20	2	11
20	11	1	9
11	9	1	2
9	2	4	1
2	1	2	0

Theorem 44: Lamé's Theorem

Let $a, m \in \mathbb{Z}^+$ with $a \geq m$. Let n be the number of divisions in Euclidian algorithm when computing $\gcd(a, m)$. Then

$$n - 1 \leq 3 \lg m$$

Proof: Lamé's Theorem

Let $r_0 = a$ and $r_1 = m$. Euclid's algorithm computes

$$\begin{aligned} r_0 &= r_1 q_1 + r_2 \qquad 0 \le r_2 < r_1 \\ r_1 &= r_2 q_2 + r_3 \qquad 0 \le r_3 < r_2 \\ &\vdots \\ r_{n-2} &= r_{n-1} q_{n-1} + r_n \qquad 0 \le r_n < r_{n-1} \\ r_{n-1} &= r_n q_n \end{aligned}$$

using n divisions to compute $r_n = \gcd(a, m)$. Note that

- $q_i \ge 1,\ i = 1, 2, \ldots, (n-1)$
- $(r_n < r_{n-1}) \Rightarrow (q_n \ge 2)$

Let F_i denote the ith Fibonacci number. Then

$$\begin{aligned} r_n &\ge 1 = F_2 \\ r_{n-1} &= r_n q_n \ge 2 r_n \ge 2 = F_3 \\ r_{n-2} &\ge r_{n-1} + r_n \ge F_3 + F_2 = F_4 \\ &\vdots \\ r_2 &\ge r_3 + r_4 \ge F_{n-1} + F_{n-2} = F_n \\ r_1 &\ge r_2 + r_3 \ge F_n + F_{n-1} = F_{n+1} \end{aligned}$$

Using the growth rate of the Fibonacci numbers $F_{n+1} \approx \phi^{n-1}$, we find

$$m = r_1 \ge F_{n+1} > \varphi^{n-1}$$

Take the logarithm base ϕ of both sides and use the change of base formula for logoarithms to derive the inequality

$$\log_\phi m = \frac{\lg m}{\lg \phi} > n - 1$$

Since $(\lg \phi)^{-1} < 3$ we have

$$3 \lg m > \frac{\lg m}{\lg \phi} > n - 1$$

Another way to state the result is that if m can be represented in k bits, then the number of divisions in Euclid's algorithm is less than 3 times the number of bits in m's binary representation.

Bézout's identity provides the link between the greatest common divisor and solving linear congruential generator.

Theorem 45: Bezout's identity

Let $a > 0$ and $m \geq 0$. Let $g = \gcd(a, m)$. Then

$$g = at + ms \quad \text{for some integers } t \text{ and } s.$$

That is, the greatest common divisor $\gcd(a, m)$ can be written as a *linear* combination of a and m.

Proof: Bezout's identity

Let $\mathbb{L} = \{ax + my > 0 : x, y \in \mathbb{Z}\}$ *be the set of all positive* linear combinations *of* a *amd* m*, and let*

$$d = \min\{ax + my > 0 : x, y \in \mathbb{Z}\}$$

be the minimum value in $\mathbb{L}$*, Let* t *and* s *be values of* x *and* y *that give the* minimum *value* d*. That is,*

$$d = at + ms > 0$$

Let a *divided by* d *give quotient* q *and remainder* r*.*

$$a = dq + r, \quad 0 \leq r < d$$

Then

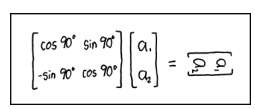
$$r = a - dq = a(1 - tq) + m(sq) \in \{ax + my \geq 0 : x, y \in \mathbb{Z}\} \text{ and } 0 \leq r < d.$$

But since d *is the smallest positive linear combination,* r *must be* 0 *and* d *divides* a*.*
A similar argument shows d *divides* m*. That is,* d *is a common divisor of* a *and* m*.*
Finally, if c *is any common divisor of* a *and* m*, then* c *divides* $d = at + ms$*. That is,* d *is the greatest common divisor of* a *and* m*.*

Extended Euclidean algorithm

Given a and m, the extension of the Euclidean algorithm computes the values of t and s from Bézout's identity

$$g = \gcd(a, m) = at + ms$$



Matrix transform

Does the determinant $xw - zy$ remind you of homogeneous coordinates and equal fractions

$$\frac{x}{z} = \frac{y}{w} \quad \text{if and only if } xw = zy.$$

Add inner (dot) products on 2 dimensional vectors and 2×2 matrix products

A little matrix algebra

Computing t and s means computing 2 values, which means a set of 2 equations. Matrix notation simplifies writing systems of equations. So, let's learn

a little bit about 2×2 matrices. The matrix

$$A = \begin{bmatrix} x & y \\ z & w \end{bmatrix} \quad \text{has determinant } xw - zy.$$

The identity matrix is

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and its determinant is} \quad \det I = 1.$$

Also, there is this result:

Lemma 1: A magic table step

Let matrix B be formed from matrix $A = \begin{bmatrix} x & y \\ z & w \end{bmatrix}$ and a quotient q using the matrix product:

$$B = \begin{bmatrix} x & y \\ z & w \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & q \end{bmatrix} = \begin{bmatrix} y & x + qy \\ w & z + qw \end{bmatrix}$$

Then

$$\begin{aligned} \det B &= y(z + qw) - w(x + qy) \\ &= (zy + qwy) - (xw + qyw) \\ &= -(xw - zy) \\ &= -\det A \end{aligned}$$

That is, the determinants have equal magnitude, but differ in sign.

Executing the extension to the Euclidean algorithm

Let's first pretend that $\gcd(a, m) = 1$. By Bézout's identity there exists integers t and s such that

$$at + ms = 1 \quad at - 1 = ms \quad \text{or } at = 1 \bmod m.$$

When $\gcd(a, m) = 1$, the numbers a and m are said to be relatively prime. Knuth uses the notation

$a \perp m$ for a and m are relatively prime.

That is,

$$t = a^{-1} \bmod m \quad \text{read "}t\text{ is the multiplicative inverse of } a \bmod m\text{."}$$

If you can compute t, then you can solve the equation below for x.

$$ax = b \bmod m \quad x = tb \bmod m$$

I've found the most simple way explain the computations of $a^{-1} = t$ is by using a *magic table*.

The magic table uses the quotients from the Euclidean algorithm to compute t and s such that

$$at + ms = \gcd(a, m)$$

Warning! I've changed my explanation from older notes. I think this computation makes the *correctness* of the procedure more obvious. It suffers from *efficiency*: All of the quotients must be computed before the extension can start. Other implementations allow the steps of the extension to coincide with the steps of the Euclidean algorithm.

Let's use the example from page 282: $\gcd(73, 17) = 1$ with quotients $\langle 4, 3, 2, 2\rangle$. Set up the magic table as below: The identity matrix on the left; the quotients, written in reverse order of their computation, above a top rule; and determinants below a bottom rule.

Let the column labeled q be the nth column for $n = 3, 4, \ldots$, and call it column_n. The values in column_n are computed using the construction in Lemma 1. That is, by the recurrence

$$\text{column}_n = q \cdot \text{column}_{(n-1)} + \text{column}_{(n-2)}$$

		2	2	3	4	Quotients
1	0					
0	1					
	+1					Determinants

$$7 \cdot 73 - 30 \cdot 17 = 511 - 510 = 1.$$

The third column, under the first quotient 2, is computed by

$$\begin{bmatrix}\text{third}\\ \text{column}\end{bmatrix} = 2 \cdot \begin{bmatrix}\text{second}\\ \text{column}\end{bmatrix} + \begin{bmatrix}\text{first}\\ \text{column}\end{bmatrix}$$

That is,

$$\det\begin{vmatrix}0 & 1\\ 1 & 2\end{vmatrix} = -1$$

$$1 = 2 \cdot 0 + 1$$
$$2 = 2 \cdot 1 + 0$$

The fourth column, under the second quotient 2, is computed by

$$\begin{bmatrix}\text{fourth}\\ \text{column}\end{bmatrix} = 2 \cdot \begin{bmatrix}\text{third}\\ \text{column}\end{bmatrix} + \begin{bmatrix}\text{second}\\ \text{column}\end{bmatrix}$$

That is,

$$\det\begin{vmatrix}1 & 2\\ 2 & 5\end{vmatrix} = +1$$

$$2 = 2 \cdot 1 + 0$$
$$5 = 2 \cdot 2 + 1$$

The fifth column, under the quotient 3, is computed by

$$\begin{bmatrix}\text{fifth}\\ \text{column}\end{bmatrix} = 3 \cdot \begin{bmatrix}\text{fourth}\\ \text{column}\end{bmatrix} + \begin{bmatrix}\text{third}\\ \text{column}\end{bmatrix}$$

That is,

$$\det\begin{vmatrix}2 & 7\\ 5 & 17\end{vmatrix} = -1$$

$$7 = 3 \cdot 2 + 1$$
$$17 = 3 \cdot 5 + 2$$

The sixth (last) column, under the quotient 4, is computed by

$$\begin{bmatrix}\text{sixth}\\ \text{column}\end{bmatrix} = 4 \cdot \begin{bmatrix}\text{fifth}\\ \text{column}\end{bmatrix} + \begin{bmatrix}\text{fourth}\\ \text{column}\end{bmatrix}$$

That is,

$$\begin{aligned} 30 &= 4 \cdot 7 + 2 \\ 73 &= 4 \cdot 17 + 5 \end{aligned}$$

The last determinant is

$$\det \begin{vmatrix} 7 & 30 \\ 17 & 73 \end{vmatrix} = 73 \cdot 7 - 17 \cdot 30 = 1$$

There are several interpretations of the equation

$$\boxed{7 \cdot 73 - 30 \cdot 17 = 1}$$

Note that

$$\begin{aligned} 7 \cdot 73 &= 511 \\ -30 \cdot 17 &= -510 \end{aligned}$$

Briefly, the equation says

$$\begin{aligned} 3 &= 5^1 \bmod 7 & 5 &= 3^1 \bmod 7 \\ 43 &= 17^{-1} \bmod 73 & 17 &= 43^{-1} \bmod 73 \\ 7 &= 13^{-1} \bmod 30 & 13 &= 7^{-1} \bmod 30 \\ 7 &= 5^{-1} \bmod 17 & 5 &= 7^{-1} \bmod 17 \end{aligned}$$

These results are shown in arithmetic detail in the text.

- Take mod 7 of both sides:

$$1 = (7 \cdot 73 - 30 \cdot 17) \bmod 7 = -30 \cdot 17 \bmod 7$$

That is, $-30 = 17^{-1}$ and $17 = (-30)^{-1} \bmod 7$. But, $17 = 3 \bmod 7$ and $-30 = 5 \bmod 7$, so that,

$$5 = 3^{-1} \quad \text{and} \quad 3 = 5^{-1} \bmod 7$$

Note that $3 \cdot 5 = 15 = 1 \bmod 7$.

- Take mod 73 of both sides:

$$1 = (7 \cdot 73 - 30 \cdot 17) \bmod 73 = -30 \cdot 17 \bmod 73$$

That is, $-30 = 17^{-1}$ and $17 = (-30)^{-1} \bmod 73$. But, $-30 = 43 \bmod 73$, so that,

$$43 = 17^{-1} \quad \text{and} \quad 17 = 43^{-1} \bmod 73$$

Note that $17 \cdot 43 = 731 = (73 \cdot 10 + 1) = 1 \bmod 73$.

- Take mod 30 of both sides:

$$1 = (7 \cdot 73 - 30 \cdot 17) \bmod 7 = 7 \cdot 73 \bmod 30$$

That is, $7 = 73^{-1}$ and $73 = 7^{-1} \bmod 30$. But, $73 = 13 \bmod 30$, so that,

$$7 = 13^{-1} \quad \text{and} \quad 13 = 7^{-1} \bmod 30$$

Note that $7 \cdot 13 = 91 = 1 \bmod 30$.

- Take mod 17 of both sides:

$$1 = (7 \cdot 73 - 30 \cdot 17) \bmod 73 = 7 \cdot 73 \bmod 17$$

That is, $7 = 73^{-1}$ and $73 = 7^{-1} \bmod 17$. But $73 = 5 \bmod 17$, so that,

$$7 = 5^{-1} \quad \text{and} \quad 5 = 7^{-1} \bmod 17$$

Note that $7 \cdot 5 = 35 == 1 \bmod 17$.

For this example

$$73 \cdot 7 - 17 \cdot 30 = +1$$

Using the information derived above, you can solve linear congruence equations. For instance,

$$\begin{aligned}
3x &= 4 \bmod 7 & x &= 3^1 \cdot 4 \bmod 7 = 5 \cdot 4 \bmod 7 = 6 \bmod 7 \\
7x &= 11 \bmod 17 & x &= 7^1 \cdot 11 \bmod 17 = 5 \cdot 11 \bmod 17 = 4 \bmod 17 \\
13x &= 8 \bmod 30 & x &= 13^1 \cdot 8 \bmod 30 = 7 \cdot 8 \bmod 30 = 26 \bmod 30 \\
17x &= 2 \bmod 73 & x &= 17^1 \cdot 2 \bmod 73 = 43 \cdot 2 \bmod 73 = 13 \bmod 73
\end{aligned}$$

Next, let's walk through the computation for example from page 282: $\gcd(92, 52) = 4$ with quotients $\langle 1, 1, 3, 3 \rangle$ and remainders $\langle 40, 12, 4, 0 \rangle$. Set up the magic table as below:

		3	3	1	1	Quotients
1	0					
0	4					
	+4					Determinants

For this example, the last determinant is

$$92 \cdot 4 - 52 \cdot 7 = +4$$

Since 92 and 54 are *not* relatively prime, equations of the form $54x = b \bmod 92$ may or may not have solutions.

Pass a Quiz: The Euclidean algorithm and its extension

Take a quiz on page 333 to check your understanding. You can return to here from the quiz.

Why does the magic table work?

I've been asked many times how the magic table works. I usually wave my hand and say: See me outside of class. No one ever does. But, to avoid humiliation, I've decided to write an explanation. Let's rename a and m by

$$a = r_{-1} \quad \text{and} \quad m = r_0$$

Then, the sequence of equations from the Euclidean algorithm is:

Dividend	=	Divisor	·	Quotient	+	Remainder
a	$=$	m	$\cdot$	q	$+$	r
r_{-1}	$=$	r_0	$\cdot$	q_1	$+$	r_1
r_0	$=$	r_1	$\cdot$	q_2	$+$	r_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
r_{k-2}	$=$	r_{k-1}	$\cdot$	q_k	$+$	r_k
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
r_{n-2}	$=$	r_{n-1}	$\cdot$	q_n	$+$	r_n

Assume the algorithm halts at the nth step when $r_n = 0$ and the last non-zero remainder is $r_{n-1} = g = \gcd(a, m)$. In this case, $r_{n-2} = gq_n$.

Look at the general equation for r_{k-2}. Write it as

$$r_{k-2} = r_k + r_{k-1} \cdot q_k$$

And consider the general magic table:

		q_n	q_{n-1}	$\cdots$	q_2	q_1	Quotients
x_n	x_{n-1}	x_{n-2}	x_{n-3}	$\cdots$	x_0	x_{-1}	
r_n	r_{n-1}	r_{n-2}	r_{n-3}	$\cdots$	r_0	r_{-1}	
	$+g$	$-g$	$+g$	$\cdots$	$(-1)^{n-1}g$	$(-1)^n g$	Determinants

The first 2×2 matrix in the magic table is

$$\begin{bmatrix} x_n & x_{n-1} \\ r_n & r_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & g \end{bmatrix} \quad \text{which has determinant } g.$$

Successive columns are computed by the recurrences

$$x_{k-2} = x_k + x_{k-1}q_k$$
$$r_{k-2} = r_k + r_{k-1}q_k$$

for $k = n, n-1, \ldots, 2, 1$.

The recurrence for the second row in the magic table computes the remainders. By definition, the second and first remainders are $r_0 = a$ and $r_m = m$. The recurrence for the first row in the magic table computes coefficients x_k and x_{k-1} such that

$$\det \begin{bmatrix} x_k & x_{k-1} \\ r_k & r_{k-1} \end{bmatrix} = (-1)^{n-k} g$$

For $k = 0$ this gives

$$\det \begin{bmatrix} x_0 & x_{-1} \\ r_0 & r_{-1} \end{bmatrix} = r_{-1}x_0 - r_0 x_{-1} = ax_0 - mx_{-1} = (-1)^n g$$

That is, the parameters t and s from from Bézout's identity

$$g = \gcd(a, m) = at + ms$$

are

- $t = -x_{-1}$ and $s = x_0$ if n is even
- $t = x_{-1}$ and $s = -x_0$ if n is odd.

Solving linear congruence equations

Now let's use these results to solve linear congruence equations. Consider

$$7x = 12 \bmod 17$$

You've computed 5 as the multiplicative inverse of 7. Therefore,

Check: $7 \cdot 9 = 63 = 3 \cdot 17 + 12 = 12 \bmod 17$.

$$x = (5 \cdot 7)x = (5 \cdot 12) \bmod 17 = 60 \bmod 17 = 9 \bmod 17$$

Change the equation to

$$7x = 12 \bmod 30$$

You know 13 is the inverse of 7 mod 30. Therefore,

Check: $7 \cdot 6 = 42 = 12 \bmod 30$.

$$x = (13 \cdot 7)x = (13 \cdot 12) \bmod 30 = 156 \bmod 30 = 6 \bmod 30$$

Here is another example: Solve $10x = 5 \bmod 37$. The steps in the Euclidean algorithm are

Dividend	=	Divisor	·	Quotient	+	Remainder
a	=	m	·	q	+	r
37	=	10	·	3	+	7
10	=	7	·	1	+	3
7	=	3	·	2	+	1
3	=	1	·	3	+	0

The magic table is

		3	2	1	3	Quotients
1	0	1	2	3	11	
0	1	3	7	10	37	
	+1	−1	+1	−1	+1	Determinants

The last two columns give the determinant

$$37 \cdot 3 - 10 \cdot 11 = 111 - 110 = 1$$

The inverse of 10 mod 37 is $-11 = 26 \bmod 37$. And, the solution of

Check: $10 \cdot 19 = 190 = (37 \cdot 5 + 5) = 5 \bmod 37$.

$$10x = 5 \bmod 37 \quad \text{is} \quad x26 \cdot 5 = 19 \bmod 37$$

Euler's totient function

The RSA encryption algorithm involves the Euler's totient function.

Recall relatively prime for a pair of integers means 1 is their greatest common divisor.

Definition 40: Euler's totient function

Let n be a positive integer. Euler's totient function $\phi(n)$ is the number of positive integers between 1 and n that are relatively prime to n. That is, $\phi(n)$ is the cardinality of the set

$$\{k : (1 \leq k \leq n) \wedge (\gcd(n, k) = 1)\}$$

You can compute $\phi(n)$ for small values of n.

$\phi(1) = 1$ since $\gcd(1,\ 1) = 1$
$\phi(2) = 1$ since $\gcd(2,\ 1) = 1$
$\phi(3) = 2$ since $\gcd(3,\ 1) = 1$ and $\gcd(3,\ 2) = 1$
$\phi(4) = 2$ since $\gcd(4,\ 1) = 1$ and $\gcd(4,\ 3) = 1$
$\phi(5) = 4$ since $\gcd(5,\ k) = 1$ for $k = 1,\ 2,\ 3,\ 4$
$\phi(6) = 2$ since $\gcd(6,\ 1) = 1$ and $\gcd(6,\ 5) = 1$

Finnish this.

Discrete logarithms

If

$$a = b^x \bmod m$$

Then x is the *(base b) discrete logarithm of a mod m*.

Include material on the discrete logarithm.

Homework Questions

Use your time outside of class to solve these problems.

1 Let $\langle r_0,\ r_1,\ r_2, \ldots \rangle$ be a sequence of *numbers* defined by the recurrence equation

$$r_n = (ar_{n-1} + b) \bmod m \quad \text{where } m = 73,\ a = 17 \text{ and } b = 37.$$

Let $r_0 = 1$ be the seed. Compute r_1 and r_2.

2 HAL is a well known Caesar cipher encoding from 2001: A Space Odyssey . Decode HAL.

3 Encode the word *discrete* using a Caesar cipher with shift 3.

4 When n eggs in a basket are removed 5 at a time none remain. When the eggs are removed 3 at a time 1 remains. When removed 4 at at time 2 remain. How many eggs are in the basket? That is, compute the value of n

that satisfies the three equations

$$n = 0 \bmod 5$$
$$n = 1 \bmod 3$$
$$n = 2 \bmod 4$$

5 Use the Euclidean algorithm to compute the greatest common divisors of the following pairs.

5.1 $\gcd(3, 5)$

5.2 $\gcd(7, 30)$

5.3 $\gcd(35, 65)$

6 Use the extended Euclidean algorithm to compute multiplicative inverses of the following numbers.

6.1 $3 \bmod 5$

6.2 $7 \bmod 30$

6.3 $35 \bmod 65$ (In this case, there is no inverse of $35 \bmod 65$.

7 Solve these linear recurrence equations. They may have one, none, or many solutions.

7.1 $3x = 4 \bmod 5$

7.2 $7x = 4 \bmod 30$

7.3 $35x = 4 \bmod 65$

8 You can determine if a number n is divisible by 2: Pretend $n = (n)_{10}$ is written in decimal. What test can you use to see if n is even? What is the test if $(n)_2$ is written in binary? What is the test if $(n)_3$ is written in base 3?

9 How can you used the digits in $n = (n)_{10}$ to determine if n has 3 as a prime factor?

10 Here are some interesting years. Why are they interesting and what are their prime factorization? What is the Euler totient value of the years?

10.1 -571 (List the prime factors of $|-571|$ and compute $\phi(571)$)

10.2 570

10.3 1054

10.4 1066

11 *Casting out nines* is a similar trick used to check arithmetic for errors. Let

$$n = (n)_{10} = \sum_{k=0}^{m-1} d_k 10^k$$

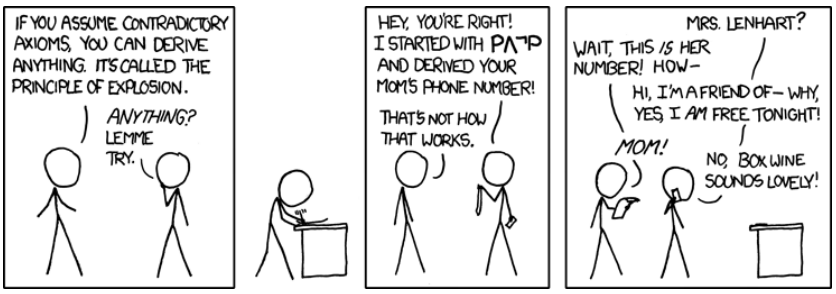
Show that

$$n \equiv \bmod 9 = \left(\sum_{k=0}^{m-1} d_k\right) \bmod 9$$

Pass a Quiz: Summative exam # 6 on time complexity and modular numbers

Take a quiz on page 334 to check your understanding. You can return to here from the quiz.

16. *Proofs by contradiction*: From *Euclid* to *Turing*



Principle of Explosion

Let P and Q be two propositions. Suppose you could show that

$$\neg\mathrm{P} \Rightarrow \mathrm{Q} \quad \text{and} \quad \neg\mathrm{P} \Rightarrow \neg\mathrm{Q}$$

were both `True`. Then you would have to conclude that $\neg\mathrm{P}$ is `False` and P is `True`. This pattern, or inference rule, is used in a proof by contradiction.

Recall the notes on Boolean logic: When an assumption P is `False`, the implication

$$\mathrm{P} \Rightarrow \mathrm{Q} = \texttt{False} \Rightarrow \mathrm{Q}$$

is `True` regardless of whether Q is `False` or `True`.

A proof by contradiction (*reductio ad absurnum*) is one method to show a propostion P is `True`. Here are the steps:

- Assume P is `False`.
- Prove that, for some proposition Q,

  $$\neg\mathrm{P} \Rightarrow \mathrm{Q} \quad \text{and} \quad \neg\mathrm{P} \Rightarrow \neg\mathrm{Q}$$

  are both `True`. But this cannot be: You've assumed P is `False`, so $\neg\mathrm{P} \Rightarrow \mathrm{Q}$ is of the form `True` $\Rightarrow \mathrm{Q}$. Similarly, $\neg\mathrm{P} \Rightarrow \neg\mathrm{Q}$ is of the form `True` $\Rightarrow \neg\mathrm{Q}$. And, one of these two conditions must be `False`. Therefore, P must be `True`.

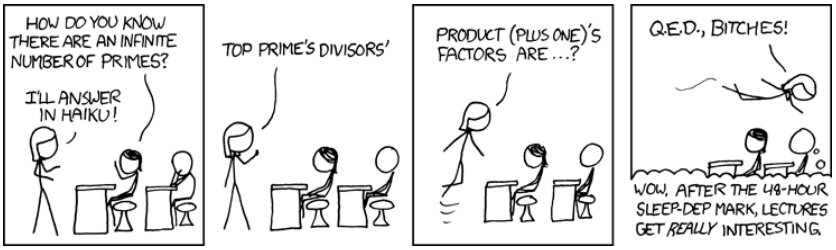
Classical Examples

In Euclid's Elements there are many famous results. Two of them are:

1. The prime numbers are unbounded.
2. $\sqrt{2}$ is not rational.

What's `True` is: For every prime p, there is another prime q larger than p.

$$(\forall p \in \mathbb{P})(\exists q \in \mathbb{P})(p < q)$$



Haiku Proof

Theorem 46: Infinitely many primes

There are infinitely many prime number.

Proof: Infinitely many primes

To prove there are infinitely many primes, assume the opposite: There are only a finite number of primes.

$$(\exists p \in \mathbb{P})(\forall q \in \mathbb{P})(q \leq p)$$

If there are only a finite number of primes, you list them:

$$p_0 < p_1 < \cdots < p_{n-1}$$

Consider the product $(p_0 p_1 \cdots p_{n-1})$ *plus* 1*, call it* q

$$q = (p_0 p_1 \cdots p_{n-1}) + 1.$$

The number q *is either prime or it is not prime.*

- *If* q *is prime, it is not on our list because it is larger than* p_{n-1}.
- *If* q *is not a prime number, then there is a prime* p *that divides* q. *But* p *cannot be any of the primes* $p_0, p_1, \ldots, p_{n-1}$ *because then* p *would divide* 1.

In both cases our list of primes in incomplete.

Next, let's establish the other classic result that $\sqrt{2}$ is irrational. Start with a corollary

Corollary 2: Even and oddness of squares

a^2 is even if and only if a is even. a^2 is odd if and only if a is odd.

Proof: Even and oddess of squares

If a *is even, then* $a = 2k$ *for some integer* k *and* $a^2 = 4k^2$ *is even. If* a *is odd, then* $a = 2k + 1$ *for some integer* k *and* $a^2 = (2k + 1)^2 =$

$4k^2 + 4k + 1$ is odd.

Now to the theorem.

Theorem 47: Irrationality of $\sqrt{2}$

The $\sqrt{2}$ is not rational.

Proof: Irrationality of $\sqrt{2}$

Assume the $\sqrt{2}$ is rational. Then $\sqrt{2}$ can be written as a common fractions

$$\sqrt{2} = \frac{a}{b} \quad \text{where } a,\ b \in \mathbb{Z} \text{ and } b \neq 0.$$

Moreover, you can assume that a/b is in "lowest terms," that is, all common factors (other than 1) have been removed so that $\gcd(a, b) = 1$.
Then follow the sequence of equations

$$\begin{aligned} \sqrt{2} &= \frac{a}{b} \\ \sqrt{2}b &= a \\ 2b^2 &= a^2 \end{aligned}$$

The last equation implies that a is even. Therefore $a^2 = 4k^2$ for some integer k and

$$\begin{aligned} 2b^2 &= 4k^2 \\ b^2 &= 2k^2 \end{aligned}$$

Which implies b is even.
The conclusion that both a and b are even contradicts the assumption that $\gcd(a, b) = 1$, and this establishes that $\sqrt{2}$ is not rational.

Modern Results

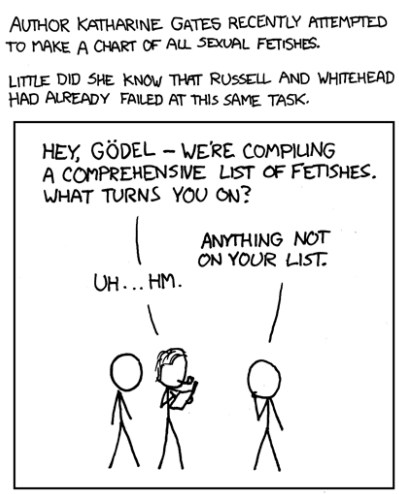
Two recently discovered concepts are:

1. Cantor proved that the real numbers are uncountable.
2. Turing showed there can be no test H that decides if program P will halt on input x.

Cantor's Diagonalization Argument

Let

$$\vec{s} = \langle s_k : s_k \in \mathbb{B},\ k \in \mathbb{N} \rangle$$



Fetishes

be a sequence of bits. For instance,

$$\vec{s} = \langle 0,\ 1,\ 1,\ 0,\ 0,\ 0,\ 1, \ldots \rangle$$

Let $\mathbb{S}$ be the set of all such sequences $\vec{s}$.

$$\mathbb{S} = \{\vec{s} = \langle s_k : s_k \in \mathbb{B},\ k \in \mathbb{N} \rangle\}$$

Note the bits along the diagonal are circled:

$$\vec{s} = \left\langle 0, 1, 0, 0, 1, 0, \ldots \right\rangle$$

The *diagonal sequence* $\vec{d}$ is computed by flipping each bit in $\vec{s}$.

Theorem 48: Cantor's diagonal argument

The set $\mathbb{S}$ of all infinite sequences of bits is *uncountable*.

Proof: Cantor's diagonal argument

Let's assume that set $\mathbb{S}$ is countable. That is, there is some one-to-one function f from the natural numbers $\mathbb{N}$ onto $\mathbb{S}$.

$$(\exists f : \mathbb{N} \to \mathbb{S})((\text{onto } f) \wedge (\text{one-to-one } f))$$

If this is the case, then f can be applied to each natural number providing list all infinite binary sequences in $\vec{S}$.
For instance, the function f might map the numbers of the sequences

$$\begin{aligned}
\vec{s_0} &= \left\langle 0, 0, 0, 0, 0, 0, \ldots \right\rangle \\
\vec{s_1} &= \left\langle 1, 1, 1, 1, 1, 1, \ldots \right\rangle \\
\vec{s_2} &= \left\langle 0, 1, 0, 0, 1, 0, \ldots \right\rangle \\
\vec{s_3} &= \left\langle 1, 0, 1, 0, 1, 1, \ldots \right\rangle \\
\vec{s_4} &= \left\langle 1, 1, 0, 0, 1, 1, \ldots \right\rangle \\
\vec{s_5} &= \left\langle 0, 0, 1, 1, 0, 0, \ldots \right\rangle \\
&\vdots \quad \vdots
\end{aligned}$$

From any enumeration, construct a sequence of bits using the values along the diagonal: Call this sequence $\vec{s}$. In our example,

$$\vec{s} = \langle 0,\ 1,\ 0,\ 0,\ 1,\ 0, \ldots \rangle$$

Next, flip each bit in s to get the "diagonal number"

$$\vec{d} = \langle 1,\ 0,\ 1,\ 1,\ 0,\ 1, \ldots \rangle$$

The sequence $\vec{d}$ cannot be in the enumerated list.

$$(\forall k \in \mathbb{N})(\vec{d} \neq \vec{s_k})$$

For each k, the k-th bit in $\vec{d}$ is the complement of the k-th bit in $\vec{s_k}$, so the two sequences cannot be equal.

Pass a Quiz: Proofs by contraction

Take a quiz on page 334 to check your understanding. You can return to here from the quiz.

Turing machines

Alan Turing was interested in the question:

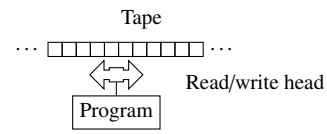
> "What can and cannot be computed?"

He devised a very simple model of an abstract computing machine. He showed it could be used to compute many things. In fact, a fundamental theory, right or wrong, underlying today's computer science is that

> "Everything computable is computable by Turing machines."

This statement is a simplified version of the Church-Turing thesis. It classifies problems: Those that have an algorithmic computable solution and those that don't.

A Turing machine:



A Turing machine is deceptively simple. Informally, a Turing machine consists of a infinitely long paper tape of cells. Each cell can hold one symbol from a finite alphabet Σ of symbols. The machine's tape head reads a symbol from the tape, considers what it knows (its *state*). Based on this information the machine decides to:

The story goes that Turing did a thought experiment: "What do I do when I compute?" He wrote the results of his experiment down and formalized it using mathematical ideas.

- Leave the tape cell unchanged or write a another symbol onto the cell.
- Stay in the same state or transition to a another state.
- Move the tape head one cell left or right.

Control of the machine is accomplished with a *transition function* that programs what the machine should do when it reads symbol s while in state q: Update the tape, update the state, move the tape head.

Basic operation are easy to program. You can multipy a binary number n by 2 by appending a 0 to the tails of n.

Decimal multiplication by 10 appends a zero, e.g., $314 \times 10 = 3140$. Binary multiplication by 2 appends a zero, e.g., $13 \times 2 = (1101)_2 \times 2 = (1\ 1010)_2 = 26$.

$$2\,(1010\ 1100)_2 = (1\ 01011000)_2 \qquad (2 \cdot 172 = 344)$$

Example: Multiply by 2

Pretend that the input tape holds a binary number, say $(1001\ 0011)_2$ (147) and you want to multiply it by 2. A Turing machine "program" that does this can be described by: Scan the tape from left-to-right copying bits until the first blank cell is found. Change the blank cell to 0 and halt.

More formally, let s be the start state and let h be the halt state. Let b be a bit or a blank $\sqcup$, and let L and R denote moving the tape head left or right. Then, the transition function for the binary multiply by 2 machine is:

$$\delta(s, b) = \begin{cases} (s, b, R) & \text{if } b \in \{0, 1\} \quad \text{copy all bits} \\ (h, 0, L) & \text{if } b = \sqcup \quad \text{append 0 after scanning all bits} \end{cases}$$

The Halting Problem

The halting problem is a decision problem: Given a program P and input x, does P halt on x?

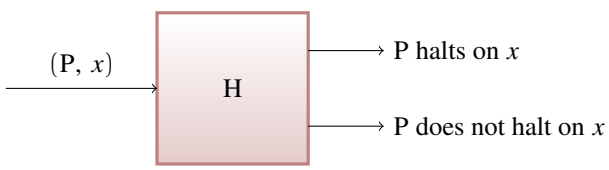
This question has a "yes" or "no" answer. The halting problem is *decidable* if there is a program H that answers the question for every program-input pair (P, x). Otherwise, the halting program is *undecidable*.

```
DEFINE DOESITHALT(PROGRAM):
{
    RETURN TRUE;
}
```

THE BIG PICTURE SOLUTION TO THE HALTING PROBLEM

Halting Problem

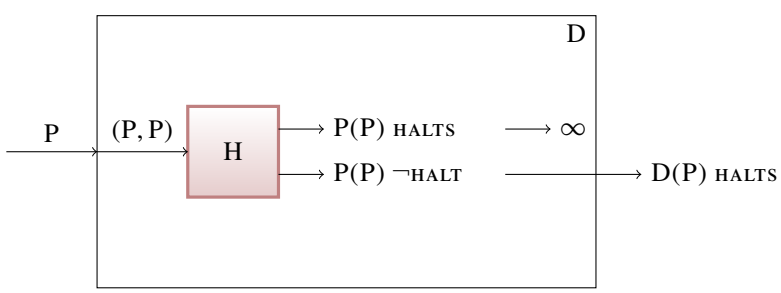
To show the halting problem is undecidable, assume the opposite: Pretend a program H exists that decides the halting problem. This assumption can be visualized as black-box H that accepts input (P, x) as input and answers "yes" or "no."



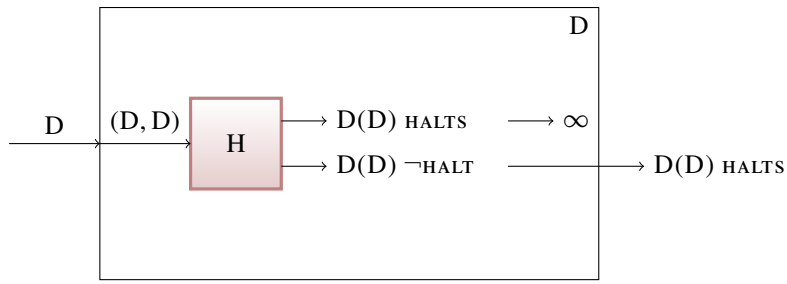
The halting machine H

If the program H exists, it can used as a subroutine other programs. Let's construct a, somewhat strange, program D. The diagonal program D:

1. Accepts a program P as input
2. Runs the halting program H on (P, P)
3. And,
   - If P(P) halts, then D(P) does not halt
   - If P(P) does not halt, then D(P) halts



Consider running program D on input D:



There are two cases:

1. If D(D) halts, then D runs to infinity.
2. If D(D) does not halt, then D halts on input D.

Both cases are contradictions. Therefore, the assumption that that a halting machine H exists must be false.

The Universal Turing machine

Explain the existence of a universal machine that can model (simulate) any other Turing machine.

The Busy Beaver Sequence

Terms in the busy beaver sequence are the maximal number of 1's that an n-state Turing machine can print on an initially blank tape before halting. Only the first few terms of the busy beaver sequence are known

n	1	2	3	4	...
β_n	1	4	6	13	...

Terms in the busy beaver sequence grow faster than any computable function.

Paradoxes

Let $\mathbb{A}$ be the set of all adjectives in the English language: Words such as *red*, *yellow*, *short*, *long polysyllabic*, *monsyllabic*, *composite*, *archaic*, *good*, *hot*, *ambiguous*, *fearful*, and so on belong in $\mathbb{A}$.

Some of these words can be used to describe themselves. For example,

short is short, so *short* is autonymous.
polysyllabic is polysyllabic, so *polysyllabic* is autonymous.

Call all adjectives that describe themselves *autonymous*. On the other hand

long is not long, so *long* is not autonymous.
monosyllabic is not monosyllabic, so *monosyllabic* is not autonymous.

Adjectives that are not autonymous are called *heteronymous*.

Now let's consider the words *autonymous* and *heteronymous*. Both are adjectives so they are either autonymous or heteronymous. The word *autonymous* literal translates as "'self-naming."

There is an important syntactic element to note in these paragraphs. An italic font is used to name a word, for example, *short* and *long*, while the normal font is used then discussing the meaning of a word. This is a distinction between the language comprised of words, and the meta-language of their meanings.

autonymous is autonymous, so *autonymous* is autonymous.

No problems here. But what about the word *heteronymous*? *Heteronymous*? translates as "different-naming." We have two possibilities

If *heteronymous* is heteronymous, then *heteronymous* is autonymous!

or

If *heteronymous* is autonymous, then *heteronymous* is heteronymous!

The above example is a form of Russell's paradox.

A set $\mathbb{X}$ is *normal* if it is not a member of itself: $\mathbb{X} \notin \mathbb{X}$.
A set $\mathbb{X}$ is *abnormal* if it is a member of itself: $\mathbb{X} \in \mathbb{X}$.

Let $\mathbb{A}$ be the set of all normal sets.

$$\mathbb{A} = \{\mathbb{X} : \mathbb{X} \notin \mathbb{X}\}$$

If $\mathbb{A}$ is normal, then $\mathbb{A} \in \mathbb{A}$ which means $\mathbb{A} \notin \mathbb{A}$. If $\mathbb{A}$ is abnormal, then $\mathbb{A} \notin \mathbb{A}$ which means $\mathbb{A} \in \mathbb{A}$.

Curry's paradox is also quite confounding. Consider the proposition P:

P = "If *this statement* is `True`, then China borders Germany."

Self-reference again: S refers to P which is a statement that involves S.

The proposition is a conditional of the form "If S, then B." Where S = "this statement is `True` " and B = "China borders German". Note that S = P,

Boolean logic tells us

- P is `True` if S = ¬P is `False`: The *statement* is `True` if the *statement* is `False`!
- P is `False` if S is `True` and B is `False`: The *statement* is `False` if the *statement* is `True`.
- P is also `True` if S is `True` and B is `True`: The statement is `True` if China borders Germany, which is `False`.

Homework Questions

Use your time outside of class to solve these problems.

1 Show that

1.1 $a^2 \equiv 0 \bmod 3$ if and only if $a \equiv 0 \bmod 3$.

1.2 $a^2 \equiv 1 \bmod 3$ if and only if $a \equiv 1 \bmod 3$ or $a \equiv 2 \bmod 3$.

1.3 $a^2 \equiv 2 \bmod 3$ has no solutions.

2 Show the $\sqrt{3}$ is not a rational number.

3 Show the $\sqrt{5}$ is not a rational number.

4 Where does the similiar proof that $\sqrt{4}$ is not a rational number fail.

5 Prove of disprove: The product of the first n primes plus 1 is a prime number. For example, $2 + 1 = 3$ is prime, $2 \cdot 3 + 1 = 7$ is prime, $2 \cdot 3 \cdot 5 + 1 = 31$ is prime, and so on.

6 Consider the (recursive) statement: "This statement has no proof" Is the statement `True` or `False`?

7 A real number r is called *sensible* if there exist positive integers a and b such that $\sqrt{a/b} = r$. For example, setting $a = 2$ and $b = 1$ shows that $\sqrt{2}$ is sensible. Prove that $\sqrt[3]{2}$ is not sensible. [§] Let a and b be positive integers, and let $g = \gcd(a, b)$.

[§] This problem came from the MIT course 6.042J/18.062J, by Prof. Albert R. Meyer and Prof. Ronitt Rubinfeld

8 Prove that $g = \gcd(a, m)$ divides the linear combination $at + ms$ for every pair of integers (t, s).

9 Prove that

$$g = \min\{as + bt > 0 : s, t \in \mathbb{Z}\}$$

That is, g is smallest positive linear combination of a and b.

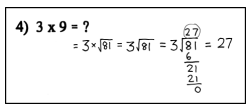
10 A *Virus tester* is a program V that accepts as input another program P and input x. If running P on x will introduce a virus, the V responds "yes." Otherwise V responds "no." Show that a virus tester cannot exist.

11 Devise a Turing machine that performs unary addition.

Quizzes

Quiz(zes) on Preliminaries

Preliminary quiz



3×9

Test your readiness for this course.

1 Evaluate each expression without using a calculator.

1.1 $\left(\frac{2}{3}\right)^{-2}$

1.2 $16^{-3/4}$

2 Simplify $3 \cdot 3^{n-1}$.

3 What is the quotient and remainder when 73 is divided by 37?

4 Simplify the algebraic expression

$$\frac{n(n-1)}{2} + n$$

5 Simplify the algebraic expression

$$(2^n - 1) + 2^n$$

6 Compute the roots of the quadratic equation

$$x^2 - x - 1 = 0$$

7 If $n = \log m$, what is an expression for m itself?

8 Convince me that each equation below is either: always `True`, always `False` or sometimes `True` and sometimes `False`.

8.1 $\sqrt{x^2 + y^2} = x + y$.

8.2 $\frac{1}{x-y} = \frac{1}{x} - \frac{1}{y}$.

8.3 $\frac{(n-1)(n-2)(n-3)}{6} + \frac{(n-1)(n-2)}{2} = \frac{n(n-1)(n-2)}{6}$.

Check your answers on page 337.
Back to the notes on page 15.

Quiz(zes) on Computers and Arithmetic

Basic computer concepts

1 Pretend the memory address register of a computer is 64-bits wide. What is the maximum size of the address space? Approximately how big is this when written in decimal notation?

2 The Internet Protocol (IP) is used to route traffic on the Internet. IPv6 uses 128-bit (16-byte) addresses.

2.1 What is the size of this address space?

2.2 Approximate your answer in decimal.

Tuesday, January 7, 2014

2.3 I just looked and the world population is 7,318,869,740 (and it grew by 92 as I wrote this). Let's just say, there are about 7.3 billion people on earth. Are there enough IPv6 addresses for them? For you?

Check your answers on page 338.
Back to the notes on page 22.

Basic number concepts

1 Using 3 digits the largest natural number that can be written is $999 = 10^3 - 1$.

Generalize: Using n digits what is the largest natural number that can be written?

2 Using 8 bits the largest natural number that can be written is $(11111111)_2 = 2^8 - 1$.

Generalize: Using n bits what is the largest natural number that can be written?

3 Generalize: Using an m numeral alphabet (e.g., $m = 10$ for decimal/digits, and $m = 2$ for binary/bits), what is the largest natural number that can be written using n numerals?

4 Pretend you can use up to 64-bits to write a number.

4.1 What is the largest number that can be written?

4.2 How many different numbers can be written?

Check your answers on page 338.
Back to the notes on page 40.

Quiz(zes) on Logic

Boolean logic basics

1 Fill in the truth tables:

Input	Output
P	$\neg$P

Input		Output
P	Q	P $\wedge$ Q

Input		Output
P	Q	P $\vee$ Q

Input		Output
P	Q	P $\Rightarrow$ Q

Input		Output
P	Q	P $\oplus$ Q

Input		Output
P	Q	P $\equiv$ Q

2 Construct a truth table for the Boolean function

$$B(\mathrm{P},\ \mathrm{Q}) = \mathrm{P} \wedge (\mathrm{P} \Rightarrow \mathrm{Q})$$

What is a more simple expression for the function?

3 Construct a truth table for the Boolean function

$$B(\mathrm{P},\ \mathrm{Q}) = \neg\mathrm{Q} \wedge (\mathrm{P} \Rightarrow \mathrm{Q})$$

What is a more simple expression for the function?

Check your answers on page 339.
Back to the notes on page 50.

Boolean functions

1 How many *truth assignments* can be made on

1.1 One Boolean variable P?

1.2 Two Boolean variables P and Q?

1.3 Three Boolean variables P, Q and R?

1.4 *n* Boolean variables?

2 How many Boolean functions can be defined on

2.1 One Boolean variable P?

2.2 Two Boolean variables P and Q?

2.3 Three Boolean variables P, Q and R?

2.4 *n* Boolean variables?

3 Given the truth table below, find a Boolean function $B(\text{P}, \text{Q}, \text{R}) = \text{Y}$ that computes it.

	Input			Output
Row	P	Q	R	Y
1	0	0	0	0
2	0	0	1	0
3	0	1	0	0
4	0	1	1	1
5	1	0	0	0
6	1	0	1	1
7	1	1	0	1
8	1	1	1	1

4 Construct a truth table to prove that

$$[(\text{P} \Rightarrow \text{Q}) \wedge (\neg\text{P} \Rightarrow \text{R})] \Rightarrow (\text{Q} \vee \text{R})$$

is a tautology: It is known as resolution.

Check your answers on page 340.
Back to the notes on page 59.

Quiz(zes) on Sets

Basics set concepts

1 Describe the following sets. List a representative sample of elements establishing a pattern. Give a function that computes the elements.

1.1 The set of even integers.

1.2 The set of odd integers.

1.3 The set of integers that have a remainder of 2 when divided by 3.

2 Both $\emptyset$ and $\{\}$ are used to denote the empty set.

2.1 Is $\emptyset = \{\emptyset\}$? That is, using the alternate notation, is $\{\} = \{\{\}\}$?

2.2 Is $\emptyset \in \{\emptyset\}$?

2.3 Is $\emptyset \in \emptyset$?

2.4 Is $\emptyset \subseteq \emptyset$?

3 Let $\mathbb{S}$ be the set of students in a class. Let $\mathbb{F}$ and $\mathbb{M}$ be the subsets of female and male students. Are the following statements `True` or `False`? Explain your answer.

Assume neither set $\mathbb{F}$ or $\mathbb{M}$ is empty.

3.1 $\mathbb{M} \in \mathbb{S}$

3.2 $\mathbb{S} \subseteq \mathbb{M}$

3.3 $\mathbb{F} \subseteq \mathbb{S}$

3.4 $\mathbb{F} \cap \mathbb{M} = \{\emptyset\}$

3.5 $\mathbb{F}$ and $\mathbb{M}$ partition $\mathbb{S}$

Check your answers on page 343.
Back to the notes on page 66.

Set operations

1 Using the three sets

$$\mathbb{X} = \{1, 2, 3, 4, 5\} \qquad \mathbb{Y} = \{0, 2, 4, 6, 8\} \qquad \mathbb{V} = \{0, 3, 5, 7, 9\}$$

over the universe of digits $\mathbb{U} = \mathbb{D} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, compute the following:

1.1 $\mathbb{X} \cup \mathbb{Y}$

1.2 $\mathbb{X} \cap \neg\mathbb{V}$

1.3 $(\mathbb{X} \cup \mathbb{Y}) \cap \mathbb{V}$

1.4 $\mathbb{X} \cup (\mathbb{Y} \cap \mathbb{V})$

1.5 Is $(\mathbb{X} \cup \mathbb{Y}) \cap \mathbb{V} = \mathbb{X} \cup (\mathbb{Y} \cap \mathbb{V})$

1.6 $\neg(\mathbb{X} \cup \mathbb{Y}) \cap \mathbb{V}$

1.7 $(\neg\mathbb{X} \cap \neg\mathbb{Y}) \cap \mathbb{V}$

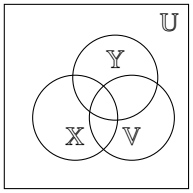
1.8 Is $\neg(\mathbb{X} \cup \mathbb{Y}) \cap \mathbb{V} = (\neg\mathbb{X} \cap \neg\mathbb{Y}) \cap \mathbb{V}$

Check your answers on page 344.
Back to the notes on page 68.

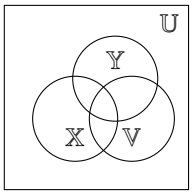
Venn diagrams

1 Shade the Venn diagram to indicate the given region.

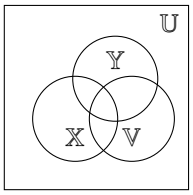
1.1 $\mathbb{X} \cap \neg\mathbb{Y}$.



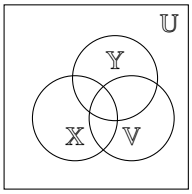
1.2 $\mathbb{X} \cup \neg\mathbb{Y}$.



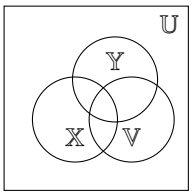
1.3 $\mathbb{X} \cap \mathbb{Y} \cap \mathbb{V}$.



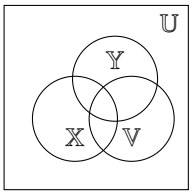
1.4 $\neg\mathbb{X} \cap \mathbb{Y} \cap \mathbb{V}$.



1.5 $(\neg\mathbb{X} \cap \neg\mathbb{Y}) \cup \neg\mathbb{V}$.



1.6 $\neg\mathbb{X} \cap (\neg\mathbb{Y} \cup \neg\mathbb{V})$.



Check your answers on page 346.
Back to the notes on page 71.

Quiz(zes) on Predicates

Basic predicate logic

1 Let $\mathbb{P} = \{2, 3, 5, 7, 11, 13, 17, 19, \ldots\}$ be the set of prime numbers. Let $p, q \in \mathbb{P}$ be prime numbers.

1.1 Is the predicate $p < q$ `True` or `False` or undecidable?

1.2 Is the quantified expression below `True` or false?

For every prime p there is a larger prime q. Or, the prime numbers are unbounded.

1.3 Negate the expression in problem (1.2).

1.4 Is the quantified expression below `True` or false? There is a prime q that larger than every other prime number.

1.5 What does the expression in problem (1.4) mean?

1.6 Negate the expression in problem (1.4).

Check your answers on page 346.
Back to the notes on page 78.

Predicates: Logic for control

1 Let x and y be integers ($x, y \in \mathbb{Z}$). Is the statement: "There is an x such that $x + y = 1$" `True` or `False` or undecidable? Explain your answer. Written mathematically the statement is:

$$(\exists\, x \in \mathbb{Z})(x + y = 1) \tag{24}$$

2 Is the statement: "For every y, there is an x such that $x + y = 1$" `True` or `False` or undecidable? Written mathematically,

$$(\forall\, y \in \mathbb{Z})(\exists\, x \in \mathbb{Z})(x + y = 1) \tag{25}$$

Explain your answer.

3 Is statement: "There is an x such that for every y, $x + y = 1$" `True` or `False` or undecidable? Written mathematically,

$$(\exists\, x \in \mathbb{Z})(\forall\, y \in \mathbb{Z})(x + y = 1) \tag{26}$$

Explain your answer.

4 Now, let's restrict x and y be natural numbers ($x, y \in \mathbb{N}$).

4.1 Is the statement: "There is an x such that $x + y = 1$" `True` or `False` or undecidable? Written mathematically,

$$(\exists\, x \in \mathbb{N})(x + y = 1) \tag{27}$$

4.2 Is the statement: "There is an x such that for every y, $x + y = 1$" `True` or `False` or undecidable? Written mathematically,

$$(\exists\, x \in \mathbb{N})(\forall\, y \in \mathbb{N})(x + y = 1) \tag{28}$$

Explain your answer.

Check your answers on page 347.
Back to the notes on page 84.

Summative exam #1

1 (10 pts) Fill in the truth table for De Morgan's first law: If P and Q are not both `True`, then one of P or Q is `False`. Conversely, if one of P or Q is `False`, then not both of them are `True`.

$$\neg(P \wedge Q) \equiv \neg P \vee \neg Q$$

De Morgan's First Law

Cases		Results	
P	Q	$\neg(P \wedge Q)$	$\neg P \vee \neg Q$

2 (10 pts) Construct a truth table for the Boolean expression.

$$[(P \to Q) \wedge (\neg P \to R)] \to (Q \vee R)$$

3 (10 pts) Given the truth table below, find a Boolean expression that computes it.

	Input			Output
Row	A	B	C	D
1	0	0	0	0
2	0	0	1	0
3	0	1	0	0
4	0	1	1	1
5	1	0	0	0
6	1	0	1	1
7	1	1	0	1
8	1	1	1	1

4 (10 pts) The Internet Protocol (IP) is used to route traffic on the Internet. IPv6 uses 128-bits (16-bytes) to name addresses. What is the size of Internet version 6 space?

5 (10 pts) Using n digits what is the largest natural number that can be written? Using n bits what is the largest natural number that can be written?

6 (10 pts) How many bits does it take to write the number 73? How many bits does it take to write the number n?

7 (30 pts) Let $n, m \in \mathbb{N}$ be natural numbers. Write the following statements using mathematical notation.

7.1 There is an $n \in \mathbb{N}$ such that $m = n/2$.

7.2 For all $m \in \mathbb{N}$, there is an $n \in \mathbb{N}$ such that $n/2 = m$.

7.3 There is an $n \in \mathbb{N}$ such that for all $m \in \mathbb{N}$, $m = n/2$.

Which of these three statements are `True`, `False`, or `undecidable`.

8 (10 pts) Okay, I have to know, who can solve my favorite quadratic equation?

$$x^2 - x - 1 = 0$$

The roots are interesting numbers, for 10 extra points, tell me about the roots.

Check your answers on page 349.
Back to the notes on page 88.

Quiz(zes) on Functions

Function Basics

1 Write, in English, an explanation of what it means to say: $f : \mathbb{X} \to \mathbb{Y}$ is a function from $\mathbb{X}$ to $\mathbb{Y}$.

2 Use predicate logic to write what it means to say: $f : \mathbb{X} \to \mathbb{Y}$ is a function from $\mathbb{X}$ to $\mathbb{Y}$.

3 Write, in English and in predicate logic, statements that describe what it means to say: $f : \mathbb{X} \to \mathbb{Y}$ is an onto function from $\mathbb{X}$ to $\mathbb{Y}$.

4 Write explanations in English and predicate logic of what it means to say $f : \mathbb{X} \to \mathbb{Y}$ is a one-to-one function.

5 Do you know your conic sections? Which conics define functions from $\mathbb{X}$ to $\mathbb{Y}$?

5.1 The parabola: $ax^2 + y = 0$

5.2 The ellipse: $\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = 1$

5.3 The hyperbola: $\left(\frac{x}{a}\right)^2 - \left(\frac{y}{b}\right)^2 = 1$

Check your answers on page 352.
Back to the notes on page 92.

The Pigeonhole Principle

Write $|\mathbb{X}| = 13$ if $\mathbb{X}$ has 13 members and say $\mathbb{X}$ has cardinality 13.

1 Suppose there are twelve people who must do 25 jobs.

1.1 Is it possible to define an onto function that maps people to jobs? If so, explain how.

1.2 Is it possible to define an onto function from jobs to people? If so, explain how.

1.3 Is it possible to define a one-to-one function from people to jobs? If so, explain how.

1.4 Is it possible to define a one-to-one function from jobs to people? If so, explain how.

2 A bridge hand is 13 cards from a standard deck of 52 cards.

2.1 Why must there be at least 4 cards of some suit in a bridge hand?

2.2 Why is there some suit with only 3 cards in a bridge hand?

Check your answers on page 353.
Back to the notes on page 98.

Examples Functions

1 Draw a graph of the following polynomial functions.

- $y = 4$
- $y = 4x - 3$
- $y = x^2 - x - 1$

2 Compute the following values.

2.1 $\lg(\sqrt[3]{16})$

2.2 $\lg(\sqrt[5]{8})$

2.3 $\lg \sqrt[3]{32}$

2.4 $\log_{16} \sqrt[3]{32}$

2.5 $\lg(2^{n!})$

3 Draw a graph of the following logarithms or exponential functions.

3.1 $y = 2^x$

3.2 $y = \lg x$

4 Identify which of the following is a permutation of the digits

$$\mathbb{D} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

For those that are, describe them using cycle notation.

4.1 $\langle 3, 4, 5, 9, 8, 7 \rangle$

4.2 $\langle 5, 4, 3, 2, 1, 0, 6, 7, 9, 8 \rangle$

4.3 $\langle 5, 5, 4, 3, 1, 1, 0, 6, 7, 8, 9 \rangle$

Check your answers on page 354.
Back to the notes on page 107.

Quiz(zes) on Horner's Rule

Horner's rule

1 Use Horner's rule to evaluate the polynomials at the indicated value of x. How many multiplications and additions were required in each case?

1.1 $p(x) = -7x^2 + 28x + 7$ at $x = 3$.

1.2 $p(x) = 4x^4 + 5x^2 - 3x + 2$ at $x = 2$.

1.3 $p(x) = 3x^5 - x^3 + 4x^2 + x + 1$ at $x = -2$.

2 How many multiplies and additions does Horner's rule make when evaluating a polynomial of degree $n - 1$?

Check your answers on page 355.
Back to the notes on page 118.

Horner's rule for conversion to decimal

Use Horner's rule to convert the following natural numbers (unsigned integers) into decimal notation.

1 $(0000\ 1001)_2$

2 $(1111\ 1111)_2$

3 $(0100\ 1100)_2$

4 $(747)_8$

5 $(123)_4$

6 $(BAD)_{16}$

Check your answers on page 356.
Back to the notes on page 121.

Remaindering to convert from decimal to another base

1 Use repeated remaindering to convert the following decimal numbers to their representation in the indicated base.

1.1 Convert 76 to binary:

1.2 Convert 137 to binary:

1.3 Convert 177 to octal:

2 Conversion between binary, octal, and hexadecimal can be done by grouping or ungrouping bits.

2.1 Group the bits you computed in question (1.1) and (1.2) three at a time to write 76 and 137 in octal.

2.2 Group the bits you computed in question (1.1) and (1.2) four at a time to write 76 and 137 in hexadecimal.

Check your answers on page 357.
Back to the notes on page 124.

Quiz(zes) on Sequences

Sequence basics

1 What is the sum of the first n terms of the Alice sequence? How are the sequence of these sums related to the Gauss sequence?

2 What is the sum of the first n terms of the Gauss sequence? How are the sequence of these sums related to the Triangular sequence?

3 Consider the sequence of even natural numbers

$$\langle 0, 2, 4, 6, 8, \ldots \rangle$$

3.1 What function $e(n)$ maps the natural numbers to the even natural numbers?

3.2 What is the recursive-initial value form for the even natural numbers?

4 Consider the sequence of natural numbers that have a remainder of 1 when divided by 3

$$\langle 1,\ 4,\ 7,\ 10,\ 13, \ldots \rangle$$

4.1 What function maps the natural numbers to these numbers?

4.2 What is the recursive-initial value form for these numbers?

5 The *half-life* sequence is the decreasing sequence

$$\left\langle 1,\ \frac{1}{2},\ \frac{1}{4},\ \frac{1}{8},\ \frac{1}{16}, \ldots \right\rangle$$

5.1 What function maps the natural numbers to these numbers?

5.2 What is the recursive-initial value form for these numbers?

Check your answers on page 359.
Back to the notes on page 132.

Differences

1 Given the sequences below, what is the function for a term in the sequence and what function computes the difference of the sequence?

1.1 The *quadratic* sequence

$$\vec{R} = \langle 0,\ 1,\ 4,\ 9,\ 16, \ldots \rangle$$

1.2 The *cubic* sequence

$$\vec{R} = \langle 0,\ 1,\ 8,\ 27,\ 64, \ldots \rangle$$

1.3 The *ratio* sequence

$$\vec{R} = \langle 1,\ r,\ r^2,\ r^3,\ r^4, \ldots \rangle$$

1.4 The *logarithm* sequence

$$\vec{L} = \langle \lg 1,\ \lg 2,\ \lg 3,\ \lg 4,\ \lg 5, \ldots \rangle$$

1.5 The *sine* sequence

$$\vec{S} = \langle \sin 0,\ \sin 1,\ \sin 2,\ \sin 3,\ \sin 4, \ldots \rangle$$

You will not be held responsible for trigonometric functions, but you may need to use them in other classes.

Check your answers on page 360.
Back to the notes on page 133.

Sums

1 Consider the general sum

$$\sum_{k=0}^{n-1} t_k$$

1.1 What is the lower limit?

1.2 What is the upper limit?

1.3 What terms are being summed?

2 What are the functions that compute the following sums?

2.1 The sum of terms in the Alice sequence

$$\sum_{k=0}^{n-1} 1$$

Note that $1 = (k+1) - k$.

2.2 The sum of terms in the Gauss sequence

$$\sum_{k=0}^{n-1} k$$

Note that $k = [(k+1)k - k(k-1)]/2$.

2.3 The sum of terms in the Doubling sequence

$$\sum_{k=0}^{n-1} 2^k$$

Note that $2^k = 2^{k+1} - 2^k$.

Check your answers on page 361.
Back to the notes on page 135.

Fundamental Theorem

1 Use the fact that

$$\triangle x^{\underline{m+1}} = (m+1)x^{\underline{m}}$$

and the fundamental theorem of the sum & difference calculus to find a simple formula for

$$\sum_{k=0}^{n-1} k^{\underline{m}}$$

2 What is a simple formula for the sum

$$\sum_{k=1}^{n} \lg\left(\frac{k+1}{k}\right)$$

Check your answers on page 362.
Back to the notes on page 136.

Recall from the notes on Horner's rule falling factorial powers are defined by

$$x^{\underline{m}} = x(x-1)(x-2)\cdots(x-m+1)$$

and it was shown that

$$\triangle x^{\underline{m}} = m x^{\underline{m-1}}$$

Sums of *ordinary* powers are more complex.

$$\sum_{k=0}^{n-1} k^0 = n$$

$$\sum_{k=0}^{n-1} k^1 = \frac{1}{2}n^2 - \frac{1}{2}n$$

$$\sum_{k=0}^{n-1} k^2 = \frac{1}{3}(n-1)^3 - \frac{2}{2}(n-1)^2 + \frac{1}{6}(n-1)$$

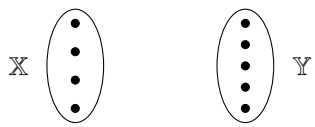
and lead to the study of Bernoulli numbers.

Summative exam #2

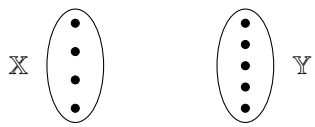
1 (10 pts) Write an English statement to explain that $f : \mathbb{X} \to \mathbb{Y}$ is an onto function. Using the notation of predicate logic, write the definition that $f : \mathbb{X} \to \mathbb{Y}$ is an onto function from $\mathbb{X}$ to $\mathbb{Y}$.

2 (20 pts) By drawing arrows from points in $\mathbb{X}$ to points in $\mathbb{Y}$ show how to construct examples of the following types, or explain why no such example can be drawn.

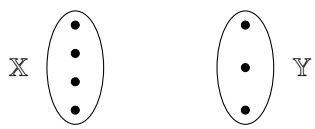
2.1 Draw a picture of a graph that is not a function.



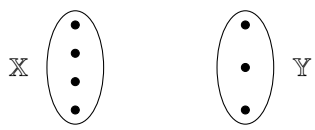
2.2 Draw a picture of a function that is onto.



2.3 Draw a picture of a function that is onto.



2.4 Draw a picture of a function that is one-to-one.



3 (10 pts) Use Horner's rule to evaluate the polynomial

$$p(x) = 3x^5 - 20x^3 - 60x - 7$$

and $x = 3$.

4 (10 pts) Use Horner's rule to convert the binary number $(1110\ 1101)_2$ to decimal notation.

5 (10 pts) Use repeated remaindering to convert the decimal number $(87)_{10}$ to binary notation.

6 (10 pts) What decimal numbers do the hexadecimal numbers A, B, C, D, E, and F represent.

7 (10 pts) Part (a) is simple arithmetic to help you guess the general answer in part (b).

7.1 Verify that the sum on the left is equal to the expression on the right.

7.1.1 $1 + 2 = \frac{2 \cdot 3}{2}$

7.1.2 $1 + 2 + 3 = \frac{3 \cdot 4}{2}$

7.1.3 $1 + 2 + 3 + 4 = \frac{4 \cdot 5}{2}$

7.2 Compute the value of the Gauss sum

$$1 + 2 + 3 + 4 + 5 + \cdots + 98 + 99 + 100$$

8 (10 pts) Part (a) is simple arithmetic to help you guess the general answer in part (b).

8.1 Verify that the sum on the left is equal to the expression on the right.

8.1.1 $1 + 2 = 2^2 - 1$

8.1.2 $1 + 2 + 4 = 2^3 - 1$

8.1.3 $1 + 2 + 4 + 8 = 2^4 - 1$

8.2 Compute the value of the Mersenne sum (you are expected to leave your answer in exponential form).

$$1 + 2 + 4 + 8 + 16 + \cdots + 2^{98} + 2^{99} + 2^{100}$$

9 (10 pts) I once gave a 20 question `True`/`False` exam.

9.1 In how many ways can you answer the questions (pretend you answer each question `True` or `False`)?

9.2 If you decide to leave some questions blank, in how many ways can you answer the questions?

Check your answers on page 364.
Back to the notes on page 139.

Quiz(zes) on Machine Numbers

Machine Number Basics

1 What is the largest unsigned natural number on:

1.1 A 16-bit computer?

1.2 A 32-bit computer?

1.3 A 64-bit computer?

2 What is the decimal value of the following ten's complement numbers? Assume the word length is 4.

2.1 $(3084)_{10c}$

2.2 $(8034)_{10c}$

3 Pad the following ten's complement numbers to be 5 digits long.

3.1 $(384)_{10c}$

3.2 $(834)_{10c}$

Check your answers on page 367.
Back to the notes on page 143.

Two's Complement Notation

1 Using two's complement notation what range of integers from most negative to most positive can be represented using

1.1 2 bits?

1.2 8 bits?

1.3 32 bits?

1.4 64 bits?

2 Negate the two's complement integers below.

2.1 $(0100\ 1100)_{2c}$.

2.2 $(1100\ 0000)_{2c}$.

2.3 $(1010\ 0100)_{2c}$.

2.4 $(1000\ 0000)_{2c}$.

3 Convert the following two's complement integers into decimal notation.

3.1 $(0100\ 1100)_{2c}$.

3.2 $(1011\ 0100)_{2c}$.

3.3 $(1010\ 1010)_{2c}$.

3.4 $(0010\ 1000)_{2c}$.

4 Convert the signed decimal integers below into two's complement notation.

4.1 $+76$.

4.2 -76.

4.3 -137.

4.4 $+177$.

Check your answers on page 367.
Back to the notes on page 151.

Floating Point Notation

1 One rule for a number m written in biased notation is that m should be a natural number. What is the minimum value for a bias b to represent the range of integers shown below?

1.1 $-127 \leq n \leq 128$.

1.2 $-511 \leq n \leq 512$.

1.3 $-49 \leq n \leq 50$.

1.4 $s \leq n \leq t$, where s and t are integers with $s \leq t$.

2 One guideline for biased notation is numbers represented should be about half-and-half positive-and-negative. Given the ranges of biased numbers below, what values of the bias b satisfies this guideline?

2.1 $0 \leq n_{\text{bias}=?} \leq 255$.

2.2 $0 \leq n_{\text{bias}=?} \leq 1023$.

2.3 $0 \leq n_{\text{bias}=?} \leq 99$.

2.4 $s \leq n_{\text{bias}=?} \leq t$, where s and t are integers with $s \leq t$.

3 Convert the biased numbers below to decimal integers.

3.1 $(6)_{\text{bias}=3}$.

3.2 $(76)_{\text{bias}=126}$.

3.3 $(137)_{\text{bias}=126}$.

3.4 $(177)_{\text{bias}=254}$.

4 The following binary strings are floating point numbers where the first (leftmost) bit is a sign bit, the next three bits are exponent bits written in biased notation with bias $b = 3$, and the last four bits are fraction bits. These floating point are normalized. What are decimal values of these floating point numbers?

4.1 $(1\ 001\ 1000)_{fp}$

4.2 $(0101\ 1011)_{fp}$

4.3 $(1111\ 1111)_{fp}$

4.4 $(0000\ 0001)_{fp}$

5 Using the 8 bit float point format, explain why $17/128$ is the smallest positive floating point number and why $16/128 = 1/8$ is not.

Check your answers on page 370.
Back to the notes on page 156.

Quiz(zes) on Names

Naming Basics

1 Using the binary alphabet, how many things can you name with fixed-length strings of the given length?

1.1 1 bit?

1.2 2 bits?

1.3 3 bits?

1.4 n bits?

2 Using the binary alphabet, how many things can you name with non-empty, variable-length strings?

2.1 1 or fewer bits?

2.2 2 or fewer bits?

2.3 3 or fewer bits?

2.4 n or fewer bits?

3 Using the English alphabet, how many things can you name with fixed-length strings of the given length?

3.1 1 character?

3.2 2 characters?

3.3 3 characters?

3.4 n characters?

4 Using the English alphabet, how many things can you name with non-empty, variable-length strings of the given length?

4.1 1 or fewer characters?

4.2 2 or fewer characters?

4.3 3 or fewer characters?

4.4 n or fewwer characters?

Check your answers on page 373.
Back to the notes on page 167.

Naming Numbers

For instance, using the binary alphabet, you can name

- 1 or 2 things using 1 bit: (0) or (0 and 1).
- 3 or 4 things using 2 bits: (00, 01, 10) or (00, 01, 10, 11)

- 5 through 8 things using 3 bits: To name the things, choose 5 through 8 elements from the set

$$\{000,\ 001,\ 010,\ 011,\ 100,\ 101,\ 110,\ 111\}$$

Using the English alphabet, you can name

- 1 through 26 things using 1 letter: a through z.
- 27 through $26^2 = 676$ things using 2 letters To name the things, choose 27 through 676 elements from the set

$$\{aa,\ ab,\ ac,\ \ldots,\ zx,\ zy,\ zz\}$$

- 677 through $26^3 = 17,576$ things using 3 letters: To name the things, choose 677 through $17,576$ elements from the set

$$\{aaa,\ aab,\ aac,\ \ldots,\ zzx,\ zzy,\ zzz\}$$

Check your answers on page 375.
Back to the notes on page 162.

Summative exam #3

1 (20 pts) The sum $1 + 2 + 4 + \ldots + 2^{n-1}$ of powers of 2 numbers called *geometric*. What function (formula) computes the value of the sum

$$1 + 2 + 4 + \ldots + 2^{n-1} = \sum_{k=0}^{n-1} 2^k$$

2 (10 pts) Convert the two's complement number $(1001\ 0100)_{2c}$ into decimal notation.

3 (10 pts) Convert the decimal integer -37 into two's complement notation.

4 Using the ideas explained in class, answer the following questions about the floating point number $(0\ 010\ 1101)_{fp}$.

4.1 (5 pts) Is the value of the number positive or negative?

4.2 (5 pts) What is the (decimal) value of the exponent?

4.3 (5 pts) What is the decimal value of the fractional part?

4.4 (5 pts) What is the (decimal) value of the normalized floating point number $(0\ 010\ 1101)_{fp}$?

5 (10 pts) The ASCII character set lies first in the Unicode alphabet: A block known as Basic Latin. The largest (last) Latin-1 value $(7F)_{16}$ represents the "delete" character. What is the decimal value of $(7F)_{16}$?

6 The octal alphabet is $\mathbb{O} = \{0, 1, 2, 3, 4, 5, 6, 7\}$.

6.1 (10 pts) How many octal numerals are needed to write 73?

6.2 (5 pts) How many octal strings with fixed-length n are there?

6.3 (5 pts) How many non-empty, variable-length octal strings of length n or less are there?

7 Consider the arithmetic sequence

$$\vec{X} = \langle 3, 7, 11, 15, \ldots, 4n-1, 4n+3, \ldots \rangle$$

and the quadratic sequence

$$\vec{Y} = \langle 0, 3, 10, 21, 36, \ldots, 2n^2 + n, \ldots \rangle$$

7.1 (5 pts) What is the difference sequence $\triangle\vec{Y}$ of $\vec{Y}$ and how is it related to $\vec{X}$?

7.2 (5 pts) What is the function (formula) that computes the sum

$$\sum_{k=0}^{n-1} (4k + 3)$$

Check your answers on page 376.
Back to the notes on page 170.

Quiz(zes) on Counting

Counting Truth Assignments and Boolean Functions

1 How many truth assignments are there on 2, 4, 8, 16 and n Boolean variables?

2 How many Boolean functions are there on 2, 4, 8, 16 and n Boolean variables?

3 In implementing Boolean logic, there are situations where we don't care what the value of an input or output is. That is, one of 3 values (`False`, `True`, and `Don't Care`) can be on an input or output line.

Some authors use 0, 1, and X for `False`, `True`, and `Don't Care`. See (Patterson and Hennessy, 1993).

3.1 For input don't cares, but only `False` and `True` output, how many Boolean functions on n variables are there?

3.2 For output don't cares, but only `False` and `True` input, how many Boolean functions on n variables are there?

3.3 For both input and output don't cares, how many Boolean functions on n variables are there?

Check your answers on page 380.
Back to the notes on page 175.

Counting Functions

1 Use theorem to show there are 2^{2^n} Boolean functions on n Boolean variables.

2 How many functions $f : \mathbb{X} \to \mathbb{Y}$ can be defined when the domain and co-domain have the following sizes?

2.1 $|\mathbb{X}| = 2$, $|\mathbb{Y}| = 4$

2.2 $|\mathbb{X}| = 4$, $|\mathbb{Y}| = 2$

2.3 $|\mathbb{X}| = 4$, $|\mathbb{Y}| = 8$

2.4 $|\mathbb{X}| = 8$, $|\mathbb{Y}| = 4$

Check your answers on page 380.
Back to the notes on page 179.

Counting Subsets

1 In how many ways can you choose 5 elements from the bits $\mathbb{B}$?

2 In how many ways can you choose 5 elements from the octal numerals $\mathbb{O}$?

3 In how many ways can you choose 5 elements from the set of digits $\mathbb{D}$?

4 In how many ways can you choose 5 elements from the set of hexadecimal digits $\mathbb{H}$?

$|\mathbb{X}| = n$

5 Let $\mathbb{X}$ be an n-element set. How many subsets does $\mathbb{X}$ have?

6 Why does the sum of values in row n of Pascal's triangle sum to 2^n?

7 Let $\mathbb{X}$ be an n-element set. Suppose there are i subsets of $\mathbb{X}$ with k-elements and j subsets of $\mathbb{X}$ with $k-1$ elements. Let y be an element that is not in $\mathbb{X}$. How many k-element subsets of $\mathbb{X} \cup \{y\}$ are there.

Check your answers on page 381.
Back to the notes on page 192.

Counting Relations

1 How many relations $x \sim y$ can be defined between elements $x \in \mathbb{X}$ and $y \in \mathbb{Y}$ when the domain and co-domain have the following sizes?

1.1 $|\mathbb{X}| = 2$, $|\mathbb{Y}| = 4$

1.2 $|\mathbb{X}| = 4$, $|\mathbb{Y}| = 2$

1.3 $|\mathbb{X}| = 4$, $|\mathbb{Y}| = 8$

1.4 $|\mathbb{X}| = 8$, $|\mathbb{Y}| = 4$

1.5 $|\mathbb{X}| = n$, $|\mathbb{Y}| = m$

2 Pretend you have 40 Facebook friends. In how many different ways can they "like" each other?

Check your answers on page 382.
Back to the notes on page 183.

Counting Perumations Cycles and Partitions

1 Use the recurrence relation

Check your answers on page 383.
Back to the notes on page 196.

Quiz(zes) on Induction

Basic induction problems

1 Write a function f that maps each integer n to some integer that has a remainder of 3 when divided by 4.

2 Use mathematical induction to prove:

$$\sum_{k=0}^{n-1} (4k+3) = n(2n+1)$$

3 Use mathematical induction to prove the largest n-numeral octal number is $8^n - 1$.

Check your answers on page 384.
Back to the notes on page 209.

Basic problems on recursion

1 Consider binary strings without consecutive 1's. For instance,

$$\epsilon,\ 0,\ 1,\ 00,\ 01,\ 10,\ 000,\ 001,\ 010,\ 100,\ 101, \ldots$$

Let be s is a string without consecutive 1's.

- how you can extend s to a string without consecutive 1 by appending a single bit.

Check your answers on page 384.
Back to the notes on page 218.

Basic problems on recursion

1 Show that the linear function $f(n) = 4n + 3$ satifies the recurrence

$$f_n = f_{n-1} + 4 \quad \text{with base case } f_0 = 3$$

2 Show that the quadratic function $s(n) = n^2$ satifies the recurrence

$$s_n = s_{n-1} + (2n - 1) \quad \text{with base case } s_0 = 0$$

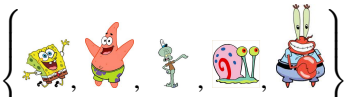
3 What well-known list of natural numbers has partial sums equal to $s(n) = n^2$?

Check your answers on page 385.
Back to the notes on page 230.

Summative exam #4

1 (10 pts) How many truth assignments and how many Boolean functions can be defined on n Boolean variables?

2 The following questions are about a set characters from the animated television series SpongeBob SquarePants. The set is called $\mathbb{SBSP}$ and the characters are: SpongeBob, Patrick, Squidward, Gary the Snail, and Captain Krabs. Let $\mathbb{SP}$ be the set without SpongeBob.

$\mathbb{SBSP} = \{$  $\}$ $\mathbb{SP} = \{$  $\}$

2.1 (10 pts) How many functions can be defined from $\mathbb{SBSP}$ to $\mathbb{SP}$?

2.2 (10 pts) Is there a one-to-one function from $\mathbb{SBSP}$ to $\mathbb{SP}$? Explain your answer.

2.3 (10 pts) Is there an onto function from $\mathbb{SBSP}$ to $\mathbb{SP}$? Explain your answer.

2.4 (10 pts) In how many ways can you arrange (permute) the characters in $\mathbb{SBSP}$?

2.5 (10 pts) How many relations can be defined between $\mathbb{SBSP}$ and $\mathbb{SP}$?

2.6 (10 pts) How many subsets does $\mathbb{SBSP}$ have?

2.7 (10 pts) How many subsets of $\mathbb{SBSP}$ contain 3 characters? Give your answer as a natural number.

3 (10 pts) Use mathematical induction to prove:

$$0 + 1 + 2 + 3 + \cdots + (n-1) = \frac{n(n-1)}{2}$$

4 (10 pts) Use mathematical induction to prove:

$$1 + 2 + 4 + \ldots + 2^{n-1} = 2^n - 1$$

Check your answers on page 387.
Back to the notes on page 241.

Quiz(zes) on Orders

Order Basics

1 Prove that divides on the natural numbers is a partial order.

2 Prove that subset on the power set of set is a partial order.

3 All other qualities being equal, would you chose to use a sorting algorithm whose running time behaved like n^2 or one that behaved like $n \lg n$? Explain your answer.

Check your answers on page 390.
Back to the notes on page 249.

Counting orders by their properties

1 How many reflexive relations can be defined on sets with cardinality 2, 8, 10, 16, and n?

2 How many antisymmetric relations can be defined on sets with cardinality 2, 8, 10, 16, and n?

Check your answers on page 390.
Back to the notes on page 253.

Quiz(zes) on Equivalences

Basics about equivalences

1 Let $\mathbb{X}$, $\mathbb{Y}$ and $\mathbb{V}$ be subsets of a universal set $\mathbb{U}$. Demonstrate the following equivalences by filling in the give truth tables.

1.1 The identity law for intersection: $\mathbb{X} \cap \mathbb{U} = \mathbb{X}$

Input		Output
$x \in \mathbb{X}$	$x \in \mathbb{U}$	$x \in \mathbb{X} \cap \mathbb{U}$
0	1	
1	1	

1.2 De Morgan's laws for the set complement of a union: $\neg(\mathbb{X} \cup \mathbb{Y}) = \neg\mathbb{X} \cap \neg\mathbb{Y}$

Input		Output	
$x \in \mathbb{X}$	$x \in \mathbb{Y}$	$x \in \neg(\mathbb{X} \cup \mathbb{Y})$	$\neg\mathbb{X} \cup \neg\mathbb{Y}$
0	0		
0	1		
1	0		
1	1		

1.3 The Currying law for sets: $(\neg\mathbb{X} \cup \mathbb{Y}) \cap (\mathbb{X} \cup \mathbb{V}) \subseteq (\mathbb{Y} \vee \mathbb{V})$

Input		Output	
$x \in \mathbb{X}$	$x \in \mathbb{Y}$	$x \in \neg(\mathbb{X} \cup \mathbb{Y})$	$\neg\mathbb{X} \cup \neg\mathbb{Y}$
0	0		
0	1		
1	0		
1	1		

2 Consider the set of integers

$$\mathbb{Z} = \{0, \pm 1, \pm 2, \pm 3, \pm 4, \pm 5, \ldots\}.$$

and the set of natural numbers

$$\mathbb{N} = \{0, 1, 2, 3, 4, 5, \ldots\}.$$

Let $a, b \in \mathbb{Z}$ and let $n \in \mathbb{N}$. If n divides $a - b$, write

$$a \equiv b \pmod{n}$$

and say a is congruent to b modula n.

3 Show that conguence modulo n is an equivalence relation.

Check your answers on page 391.
Back to the notes on page 262.

Summative exam #5

1 (10 pts) Show that the function $t(n) = n(n-1)/2$ satisfies the recurrence equation and initial condition

$$t_n = t_{n-1} + (n-1), \quad t_0 = 0$$

2 (10 pts) Prove that $t(n) = n \lg n$ satisfies the recurrence

$$t_{2n} = 2t_n + 2n, \; t_1 = 0, \quad \text{for } n = 1, 2, 4, 8, \ldots$$

3 (35 pts) Consider the divides relation $(a \mid b)$ on the set of natural numbers.

3.1 Does 7 divide 38? Explain your answer.

3.2 Is divides reflexive?

3.3 Is divides antisymmetric? Explain your answer.

3.4 Is divides symmetric? Explain your answer.

3.5 Is divides transitive? Explain your answer.

3.6 Is divides an equivalence? Explain your answer.

3.7 Is divides a partial order? Explain your answer.

4 (35 pts) Consider congruence mod m relation ($a \equiv b \pmod{m}$) on the set of integers.

4.1 What does it mean to say that a is congruent to b mod m?

4.2 Is $a \equiv a \pmod{m}$ for every integer a? Explain your answer.

4.3 If $a \equiv b \pmod{m}$ is $b \equiv a \pmod{m}$ for every pair of integers a and b? Explain your answer.

4.4 If $a \equiv b \pmod{m}$ and $b \equiv a \pmod{m}$ does $a = b$ for every pair of integers a and b? Explain your answer.

4.5 If $a \equiv b \pmod{m}$ and $b \equiv c \pmod{m}$ is $a \equiv c \pmod{m}$ for every triple of integers a, b and c? Explain your answer.

4.6 Is congruence mod m an equivalence? Explain your answer.

4.7 Is congruence mod m a partial order? Explain your answer.

5 (10 pts) What function $f(n)$ satisfies the recurrence $f_n = 2f_{n-1} + 1$ and initial condition $f_0 = 0$? Listing the first few terms generated by the recurrence should help you identify $f(n)$.

Check your answers on page 392.
Back to the notes on page 271.

Quiz(zes) on Modular Numbers

Basic cryptography concepts

1 Consider the modular recurrence equation

$$r_n = (4r_{n-1} + 2) \bmod 7.$$

with seed $r_0 = 1$.

1.1 What sequence is generated?

1.2 What is the *period* of the sequence?

2 Pretend you use a Caesar cipher with shift $k = 6$ to send and receive messages.

2.1 What is the encryption of "mrwatsoncomehere?"

2.2 What is decryption of the message "'igxvkjoks?"

3 Pretend you use an affine cipher

$$(5x + 7) \bmod 26$$

to send and receive messages. Encrypt the message "soitgoes"

"So it goes" is a frequent refrain in Kurt Vonnegut's novel, Slaughterhouse-Five, or The Children's Crusade: A Duty-Dance with Death

Check your answers on page 396.
Back to the notes on page 276.

Modular arithmetic

1 Verify the following:

1.1 Verify that 2 is the remainder when 37 is divided by 7. Conclude that 37 can be reduced to 2 mod 7: $37 = 2 \bmod 2$.

1.2 Verify that 5 is the (smallest non-negative) remainder when -37 is divided by 7. Conclude that -37 can be reduced to 5 mod 7.

1.3 Do you see how you can use $-37 + 37 = 0$ together with $37 = 2 \bmod 7$ to deduce that $-37 = 5 \bmod 7$?

2 Verify the given value of x satisfies the linear congruence.

2.1 $x = 5$ satisfies $3x = 1 \bmod 7$, that is $3^{-1} = 5 \bmod 7$.

2.2 $x = 5$ satisfies $5x = 1 \bmod 8$, that is $5^{-1} = 5 \bmod 8$.

Check your answers on page 396.
Back to the notes on page 280.

The Euclidean algorithm and its extension

1 Given the factorization of numbers below, identify their greatest common divisor.

1.1 $a = 210 = 2 \cdot 3 \cdot 5 \cdot 7$ and $m = 495 = 3^2 \cdot 5 \cdot 11$

1.2 $a = 147 = 3 \cdot 7^2$ and $m = 392 = 2^3 \cdot 7^2$

2 Use the Euclidean algorithm to compute the greatest common divisor of the numbers below. Identify the sequence of quotients and remainders that are generated.

2.1 $\gcd(m, a) = \gcd(37, 17)$

2.2 $\gcd(m, a) = \gcd(165, 42)$

Check your answers on page 397.
Back to the notes on page 288.

Quiz(zes) on Proofs

Proofs by contradiction

1 Mimic Euclid's proof that $\sqrt{2}$ is *irrational* to show that $\sqrt{3}$ is *irrational*.

2 Prove there is no largest composite number.

Check your answers on page 398.
Back to the notes on page 299.

Summative exam #6

1 (12 pts) Consider the modular recurrence equation

$$r_n = (4r_{n-1} + 2) \bmod 9.$$

with seed $r_0 = 0$. What are the values of r_1, r_2, r_3?

2 (12 pts) Consider -37 divided by 5.

2.1 What is the quotient q and smallest non-negative remainder r?

2.2 Reduce $-37 \bmod 5$ and $37 \bmod 5$ to modular integers in $\mathbb{Z}_5$.

3 (10 pts) Verify the following.

3.1 Show that $x = 4$ satisfies $3x = 1 \bmod 11$.

3.2 What is $3^{-1} \bmod 11$?

4 (12 pts) Use the Euclidean algorithm to compute the greatest common divisor of 19 and 43.

5 (12 pts) Find parameters s and t such that

$$43s + 19t = \gcd(43, 19)$$

6 (10 pts) Compute the value of x that solves the linear congruence equation

$$19x = 4 \bmod 43$$

7 (12 pts) Use a proof by contradiction to show that $\sqrt[3]{2}$, the cube root of 2, is irrational.

8 (10 pts) What function $f(n)$ satisfies the recurrence $f_n = f_{n-1} + (n-1)$ and initial condition $f_0 = 0$? Listing the first few terms generated by the recurrence should help you identify $f(n)$.

9 (10 pts) Let $\mathbb{X} = \{a, b, c, d\}$.

9.1 Is $\{\{a, b\}, \{c, d\}\}$ a partition of $\mathbb{X}$? Explain your answer.

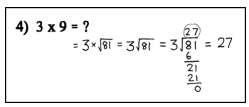
9.2 Is $\{\{a, b, c\}, \{c, d\}\}$ a partition of $\mathbb{X}$? Explain your answer.

Check your answers on page 398.
Back to the notes on page 293.

Keys

Keys to Quiz(zes) on Preliminaries

Preliminary quiz (Key)



3×9

Test your readiness for this course.

Answer: I can't give answers because I need to test if you are ready.

1 Evaluate each expression without using a calculator.

1.1 $\left(\frac{2}{3}\right)^{-2}$

1.2 $16^{-3/4}$

2 Simplify $3 \cdot 3^{n-1}$.

3 What is the quotient and remainder when 73 is divided by 37?

4 Simplify the algebraic expression

$$\frac{n(n-1)}{2} + n$$

5 Simplify the algebraic expression

$$(2^n - 1) + 2^n$$

6 Compute the roots of the quadratic equation

$$x^2 - x - 1 = 0$$

7 If $n = \log m$, what is an expression for m itself?

8 Convince me that each equation below is either: always `True`, always `False` or sometimes `True` and sometimes `False`.

8.1 $\sqrt{x^2 + y^2} = x + y$.

8.2 $\frac{1}{x-y} = \frac{1}{x} - \frac{1}{y}$.

8.3 $\frac{(n-1)(n-2)(n-3)}{6} + \frac{(n-1)(n-2)}{2} = \frac{n(n-1)(n-2)}{6}$.

Back to the quiz on page 305.
Back to the notes on page 15.

Keys to Quiz(zes) on Computers and Arithmetic

Basic computer concepts (Key)

1 Pretend the memory address register of a computer is 64-bits wide. What is the maximum size of the address space? Approximately how big is this when written in decimal notation?

Answer: The size of the address space is $2^{64} \approx 10^{19.2}$ about 18 quintillion words. Note the operating system may partition memory restricting addressable memory. Well, at least, that is a goal of security.

$$\begin{aligned} 2^{64} &= 18,446,744,073,709,551,616 \\ &\approx 1.845 \times 10^{19} \\ 10^{19.2} &\approx 1.585 \times 10^{19} \end{aligned}$$

2 The Internet Protocol (IP) is used to route traffic on the Internet. IPv6 uses 128-bit (16-byte) addresses.

2.1 What is the size of this address space?

Answer: The Internet space has 2^{128} addresses.

2.2 Approximate your answer in decimal.

Answer:

$$2^{128} = (2^{10})^{12.8} \approx (10^3)^{12.8} = 10^{38.4}$$

Tuesday, January 7, 2014

2.3 I just looked and the world population is 7,318,869,740 (and it grew by 92 as I wrote this). Let's just say, there are about 7.3 billion people on earth. Are there enough IPv6 addresses for them? For you?

Answer: There are about

$$\frac{10^{38.4}}{7.3 \times 10^9} = \frac{10^{29.4}}{7.3} \approx 1.37 \times 10^{28.4}$$

IPv6 addresses per person. That seems like enough for me.

Back to the quiz on page 306.
Back to the notes on page 22.

Basic number concepts (Key)

1 Using 3 digits the largest natural number that can be written is $999 = 10^3 - 1$.

Generalize: Using n digits what is the largest natural number that can be written?

Answer: The largest decimal number that can written using n digits is $10^n - 1$.

2 Using 8 bits the largest natural number that can be written is $(11111111)_2 = 2^8 - 1$.

Generalize: Using n bits what is the largest natural number that can be written?

Answer: The largest binary number that can written using n bits is $2^n - 1$.

3 Generalize: Using an m numeral alphabet (e.g., $m = 10$ for decimal/digits, and $m = 2$ for binary/bits), what is the largest natural number that can be written using n numerals?

Answer: The largest n numeral number that can written using an m numeral alphabet is $m^n - 1$.

4 Pretend you can use up to 64-bits to write a number.

4.1 What is the largest number that can be written?

Answer: The largest number that can be written is a sequence of 64 bits all set to 1: This represents the decimal number

$$2^{64} - 1 = 18,446,744,073,709,551,615 \approx 1.8 \times 10^{19}$$

4.2 How many different numbers can be written?

Answer: There are

$$2^{64} = 18,446,744,073,709,551,616$$

different numbers that can be written.

Back to the quiz on page 306.
Back to the notes on page 40.

Keys to Quiz(zes) on Logic

Boolean logic basics (Key)

1 Fill in the truth tables: Answer:

Input P	Output $\neg$P
0	1
1	0

Input P	Input Q	Output P $\wedge$ Q
0	0	0
0	1	0
1	0	0
1	1	1

Input P	Input Q	Output P $\vee$ Q
0	0	0
0	1	1
1	0	1
1	1	1

Input P	Input Q	Output P $\Rightarrow$ Q
0	0	1
0	1	1
1	0	0
1	1	1

Input P	Input Q	Output P $\oplus$ Q
0	0	0
0	1	1
1	0	1
1	1	0

Input P	Input Q	Output P $\equiv$ Q
0	0	1
0	1	0
1	0	0
1	1	1

2 Construct a truth table for the Boolean function

$$B(\mathrm{P}, \mathrm{Q}) = \mathrm{P} \wedge (\mathrm{P} \Rightarrow \mathrm{Q})$$

What is a more simple expression for the function?

Answer: The column in crimson displays the output.

Input		Output		
P	Q	P	$\wedge$	$(\mathrm{P} \Rightarrow \mathrm{Q})$
0	0	0	0	1
0	1	0	0	1
1	0	1	0	0
1	1	1	1	1

The expression $\mathrm{P} \wedge (\mathrm{P} \Rightarrow \mathrm{Q})$ is equivalent (*can be reduced*) to $\mathrm{P} \wedge \mathrm{Q}$.

3 Construct a truth table for the Boolean function

$$B(\mathrm{P}, \mathrm{Q}) = \neg\mathrm{Q} \wedge (\mathrm{P} \Rightarrow \mathrm{Q})$$

What is a more simple expression for the function?

Answer: The column in crimson displays the output.

Input		Output		
P	Q	$\neg$Q	$\wedge$	$(\mathrm{P} \Rightarrow \mathrm{Q})$
0	0	1	1	1
0	1	0	0	1
1	0	1	0	0
1	1	0	0	1

The function is equivalent (*can be reduced*) to $\neg(\mathrm{P} \vee \mathrm{Q}) \equiv \neg\mathrm{P} \wedge \neg\mathrm{Q}$.

Back to the quiz on page 307.
Back to the notes on page 50.

Boolean functions (Key)

1 How many *truth assignments* can be made on

1.1 One Boolean variable P?

Answer: There are $2 = 2^1$ truth assignments: `False` and `True`.

1.2 Two Boolean variables P and Q?

Answer: There are $4 = 2^2$ truth assignments:

(`False`, `False`), (`False`, `True`), (`True`, `False`), (`True`, `True`)

1.3 Three Boolean variables P, Q and R?

Answer: There are $8 = 2^3$ truth assignments.

1.4 n Boolean variables?

Answer: There are 2^n truth assignments.

2 How many Boolean functions can be defined on

2.1 One Boolean variable P?

Answer: There are $4 = 2^{2^1}$ Boolean functions

1. $B(\mathrm{P}) = \texttt{False}$
2. $B(\mathrm{P}) = \texttt{True}$
3. $B(\mathrm{P}) = \mathrm{P}$
4. $B(\mathrm{P}) = \neg\mathrm{P}$

2.2 Two Boolean variables P and Q?

Answer: There are 16 different Boolean functions (expressions) on two variables.

There are four basic conjunctions: $\mathrm{P} \wedge \mathrm{Q}$, $\mathrm{P} \wedge \neg\mathrm{Q}$, $\neg\mathrm{P} \wedge \mathrm{Q}$, $\neg\mathrm{P} \wedge \neg\mathrm{Q}$. They can be `Or`-ed together to create 16 different functions.

1. There are the $4 = \binom{4}{1}$ basic conjunctions.
2. There are $6 = \binom{4}{2}$ different ways to `Or` the basic conjunctions together.
   (a) $(\mathrm{P} \wedge \mathrm{Q}) \vee (\mathrm{P} \wedge \neg\mathrm{Q}) \mapsto \mathrm{P}$
   (b) $(\mathrm{P} \wedge \mathrm{Q}) \vee (\neg\mathrm{P} \wedge \mathrm{Q}) \mapsto \mathrm{Q}$
   (c) $(\mathrm{P} \wedge \mathrm{Q}) \vee (\neg\mathrm{P} \wedge \neg\mathrm{Q}) \mapsto \mathrm{P} \equiv \mathrm{Q}$
   (d) $(\mathrm{P} \wedge \neg\mathrm{Q}) \vee (\neg\mathrm{P} \wedge \mathrm{Q}) \mapsto \mathrm{P} \oplus \mathrm{Q}$
   (e) $(\mathrm{P} \wedge \neg\mathrm{Q}) \vee (\neg\mathrm{P} \wedge \neg\mathrm{Q}) \mapsto \neg\mathrm{Q}$
   (f) $(\neg\mathrm{P} \wedge \mathrm{Q}) \vee (\neg\mathrm{P} \wedge \neg\mathrm{Q}) \mapsto \neg\mathrm{P}$
3. There are $4 = \binom{4}{3}$ different ways to `Or` three of the basic conjunctions together. You can verify these four functions are
   (a) $\neg\mathrm{P} \vee \mathrm{Q}$, which can be written as $\mathrm{P} \Rightarrow \mathrm{Q}$
   (b) $\mathrm{P} \vee \neg\mathrm{Q}$, which can be written as $\mathrm{Q} \Rightarrow \mathrm{P}$
   (c) $\mathrm{P} \vee \mathrm{Q}$, which can be written as $\neg\mathrm{P} \Rightarrow \mathrm{Q}$, or its contrapositive $\neg\mathrm{Q} \Rightarrow \mathrm{P}$
   (d) $\neg\mathrm{P} \vee \neg\mathrm{Q}$, which can be written as $\mathrm{P} \Rightarrow \neg\mathrm{Q}$, or its contrapositive $\mathrm{Q} \Rightarrow \neg\mathrm{P}$
4. And, there is $1 = \binom{4}{4}$ way to `Or` all four of the basic conjunctions together. The result can be reduced to `True`.

But this is only $4 + 6 + 4 + 1 = 15$ different functions (expressions). There were supposed to be 16 expressions! The missing one is `False`, which can counted by not selecting any of the basic conjunctions, and there is $1 = \binom{4}{0}$ way to do this.

2.3 Three Boolean variables P, Q and R?

Answer: Generalize: There are $8 = 2^3$ truth assignments on 3 Boolean variables. Each truth assignment can be mapped to one of two values: `True` or `False`. Therefore, are $256 = 2^8 = 2^{2^3}$ three variable Boolean functions.

2.4 n Boolean variables?

Answer: There are 2^{2^n} n variable Boolean functions.

3 Given the truth table below, find a Boolean function $B(\text{P}, \text{Q}, \text{R}) = \text{Y}$ that computes it.

	Input			Output
Row	P	Q	R	Y
1	0	0	0	0
2	0	0	1	0
3	0	1	0	0
4	0	1	1	1
5	1	0	0	0
6	1	0	1	1
7	1	1	0	1
8	1	1	1	1

Answer: Taking the row where the output Y is 1 form AND-clauses of the input to make the row `True`.

$$\text{Y} = (\neg\text{P} \wedge \text{Q} \wedge \text{R}) \vee (\text{P} \wedge \neg\text{Q} \wedge \text{R}) \vee (\text{P} \wedge \text{Q} \wedge \neg\text{R}) \vee (\text{P} \wedge \text{Q} \wedge \text{R})$$

Alternatively, take the row where the output Y is 0 form OR-clauses of the input to make the row `False`.

$$\text{Y} = (\text{P} \vee \text{Q} \vee \text{R}) \wedge (\text{P} \vee \text{Q} \vee \neg\text{R}) \wedge (\text{P} \vee \neg\text{Q} \vee \text{R}) \wedge (\neg\text{P} \vee \text{Q} \vee \text{R})$$

4 Construct a truth table to prove that

$$[(\text{P} \Rightarrow \text{Q}) \wedge (\neg\text{P} \Rightarrow \text{R})] \Rightarrow (\text{Q} \vee \text{R})$$

is a tautology: It is known as resolution.

Answer:

Input					Output		
P	Q	R	$(\text{P} \Rightarrow propq)$	$\wedge$	$(\neg\text{P} \Rightarrow \text{R})$	$\Rightarrow$	$(\text{Q} \vee \text{R})$
0	0	0	1	0	0	1	0
0	0	1	1	1	1	1	1
0	1	0	1	0	0	1	1
0	1	1	1	1	1	1	1
1	0	0	0	0	1	1	0
1	0	1	0	0	1	1	1
1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1

Back to the quiz on page 307.
Back to the notes on page 59.

Keys to Quiz(zes) on Sets

Basics set concepts (Key)

1 Describe the following sets. List a representative sample of elements establishing a pattern. Give a function that computes the elements.

1.1 The set of even integers.

Answer: The set of even integers is

$$\{\ldots, -4, -2, 0, 2, 4, \ldots\}$$

Common names for the even integers are $2\mathbb{Z}$, $[0]_2$ or $0 \pmod 2$. The function that computes the elements is

$$f : \mathbb{Z} \to \mathbb{Z} \quad \text{where } f(n) = 2n.$$

1.2 The set of odd integers.

Answer: The set of odd integers is

$$\{\ldots, -5, -3, -1, 1, 3, 5, \ldots\}$$

Common names for the odd integers are $[1]_2$ or $1 \pmod 2$. The function that computes the elements is

$$f : \mathbb{Z} \to \mathbb{Z} \quad \text{where } f(n) = 2n + 1.$$

1.3 The set of integers that have a remainder of 2 when divided by 3.

Answer: The remainder 2 on divide by 3 set is

$$\{\ldots, -7, -4, -1, 2, 5, 8, \ldots\}$$

Common names for these integers are $[2]_3$ and $2 \pmod 3$. The function that computes the elements is

$$f : \mathbb{Z} \to \mathbb{Z} \quad \text{where } f(n) = 3n + 2.$$

2 Both $\emptyset$ and $\{\}$ are used to denote the empty set.

2.1 Is $\emptyset = \{\emptyset\}$? That is, using the alternate notation, is $\{\} = \{\{\}\}$?

Answer: No, the empty set is not the same as the set containing the empty set.

2.2 Is $\emptyset \in \{\emptyset\}$?

Answer: Yes, the empty set is a member of the set containing the empty set.

2.3 Is $\emptyset \in \emptyset$?

Answer: No, the empty set is not a member of itself.

2.4 Is $\emptyset \subseteq \emptyset$?

Answer: Yes, Every element in the empty set is a member of the empty set: The statement $x \in \emptyset \Rightarrow x \in \emptyset$ is `True` because the assumption $x \in \emptyset$ is `False`.

3 Let $\mathbb{S}$ be the set of students in a class. Let $\mathbb{F}$ and $\mathbb{M}$ be the subsets of female and male students. Are the following statements `True` or `False`? Explain your answer.

Assume neither set $\mathbb{F}$ or $\mathbb{M}$ is empty.

3.1 $\mathbb{M} \in \mathbb{S}$

Answer: `False`, The symbol $\in$ means "is a member of." $\mathbb{M}$ is subset of $\mathbb{S}$ ($\mathbb{M} \subseteq \mathbb{S}$), but it is *not an element of* $\mathbb{S}$.

3.2 $\mathbb{S} \subseteq \mathbb{M}$

Answer: In general this would be `False`. It would only be `True` if there were no female students in the class.

3.3 $\mathbb{F} \subseteq \mathbb{S}$

Answer: This is `True`: Every female student is a student.

3.4 $\mathbb{F} \cap \mathbb{M} = \{\emptyset\}$

Answer: `False`, the intersection is $\emptyset$, not the set containing the empty set.

3.5 $\mathbb{F}$ and $\mathbb{M}$ partition $\mathbb{S}$

Answer: A partition cuts $\mathbb{S}$ up into disjoint subsets that cover $\mathbb{S}$. Assigning gender is a biological/medical decision. Under the common interpretation female/male would partition the set of students.

Back to the quiz on page 308.
Back to the notes on page 66.

Set operations (Key)

1 Using the three sets

$$\mathbb{X} = \{1, 2, 3, 4, 5\} \qquad \mathbb{Y} = \{0, 2, 4, 6, 8\} \qquad \mathbb{V} = \{0, 3, 5, 7, 9\}$$

over the universe of digits $\mathbb{U} = \mathbb{D} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, compute the following:

1.1 $\mathbb{X} \cup \mathbb{Y}$

Answer: $\mathbb{X} \cup \mathbb{Y} = \{0, 1, 2, 3, 4, 5, 6, 8\}$

1.2 $\mathbb{X} \cap \neg\mathbb{V}$

Answer: $\mathbb{X} \cap \neg\mathbb{V} = \{1, 2, 4\}$

1.3 $(\mathbb{X} \cup \mathbb{Y}) \cap \mathbb{V}$

Answer: $(\mathbb{X} \cup \mathbb{Y}) \cap \mathbb{V} = \{0, 3, 5\}$

1.4 $\mathbb{X} \cup (\mathbb{Y} \cap \mathbb{V})$

Answer: $\mathbb{X} \cup (\mathbb{Y} \cap \mathbb{V}) = \{0, 1, 2, 3, 4, 5\}$

1.5 Is $(\mathbb{X} \cup \mathbb{Y}) \cap \mathbb{V} = \mathbb{X} \cup (\mathbb{Y} \cap \mathbb{V})$

Answer: Well no, as you see if you computed the two expressions correctly. The upshot is: The order in which set operations matters, just as the order of addition and multiplication matters in arithmetic.

1.6 $\neg(\mathbb{X} \cup \mathbb{Y}) \cap \mathbb{V}$

Answer: $\neg(\mathbb{X} \cup \mathbb{Y}) \cap \mathbb{V} = \{7, 9\}$

1.7 $(\neg\mathbb{X} \cap \neg\mathbb{Y}) \cap \mathbb{V}$

Answer: $(\neg\mathbb{X} \cap \neg\mathbb{Y}) \cap \mathbb{V} = \{7, 9\}$

1.8 Is $\neg(\mathbb{X} \cup \mathbb{Y}) \cap \mathbb{V} = (\neg\mathbb{X} \cap \neg\mathbb{Y}) \cap \mathbb{V}$

Answer: Well yes, this is an application of De Morgan's laws.

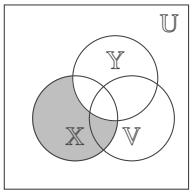
Back to the quiz on page 309.
Back to the notes on page 68.

Venn diagrams (Key)

1 Shade the Venn diagram to indicate the given region.

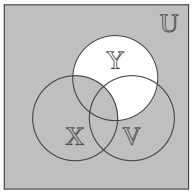
1.1 $\mathbb{X} \cap \neg\mathbb{Y}$.

Answer:



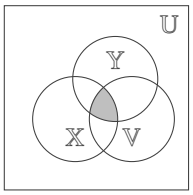
1.2 $\mathbb{X} \cup \neg\mathbb{Y}$.

Answer:



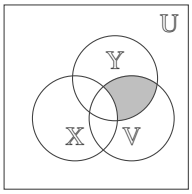
1.3 $\mathbb{X} \cap \mathbb{Y} \cap \mathbb{V}$.

Answer:



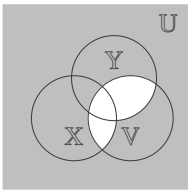
1.4 $\neg\mathbb{X} \cap \mathbb{Y} \cap \mathbb{V}$.

Answer:



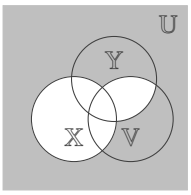
1.5 $(\neg\mathbb{X} \cap \neg\mathbb{Y}) \cup \neg\mathbb{V}$.

Answer:



1.6 $\neg\mathbb{X} \cap (\neg\mathbb{Y} \cup \neg\mathbb{V})$.

Answer:



Back to the quiz on page 310.
Back to the notes on page 71.

Keys to Quiz(zes) on Predicates

Basic predicate logic (Key)

1 Let $\mathbb{P} = \{2, 3, 5, 7, 11, 13, 17, 19, \ldots\}$ be the set of prime numbers. Let $p, q \in \mathbb{P}$ be prime numbers.

1.1 Is the predicate $p < q$ `True` or `False` or undecidable?

Answer: The inequality is undeciable: Without known values for p and q, you cannot decide if the predicate is `True` or `False`.

1.2 Is the quantified expression below `True` or false?

For every prime p there is a larger prime q. Or, the prime numbers are unbounded.

Answer: It is `True`. For every prime p there is a larger prime q. Name a prime number. Eventually you will be able to name another larger prime number.

1.3 Negate the expression in problem (1.2).

Answer: There is largest prime number. There does exist a prime q that is larger than every other prime number p.

1.4 Is the quantified expression below `True` or false? There is a prime q that larger than every other prime number.

Answer: This is `False`. Euclid's Elements contains a proof that there the primes are unbounded. The proof is be contradiction. The assumption that the primes are finite leads to a falsehood.

1.5 What does the expression in problem (1.4) mean?

Answer: The expression says there is some prime q that is larger than every other prime number p.

1.6 Negate the expression in problem (1.4).

Answer:

$$(\forall\, q \in \mathbb{P})(\exists\, p \in \mathbb{P})(p > q)$$

Back to the quiz on page 310.
Back to the notes on page 78.

Predicates: Logic for control (Key)

1 Let x and y be integers ($x, y \in \mathbb{Z}$). Is the statement: "There is an x such that $x + y = 1$" `True` or `False` or undecidable? Explain your answer. Written mathematically the statement is:

$$(\exists\, x \in \mathbb{Z})(x + y = 1) \tag{29}$$

Answer: The statement is `True`. The value of y is not specified, but whatever it is, if you set $x = (1 - y)$, then $x + y = (1 - y) + y = 1$ will be `True`. Notice also that if y is an integer, then $x = 1 - y$ is also an integer.

2 Is the statement: "For every y, there is an x such that $x + y = 1$" `True` or `False` or undecidable? Written mathematically,

$$(\forall\, y \in \mathbb{Z})(\exists\, x \in \mathbb{Z})(x + y = 1) \tag{30}$$

Explain your answer.

Answer: The statement is `True` by the reasoning in the answer to question (1).

3 Is statement: "There is an x such that for every y, $x + y = 1$" `True` or `False` or undecidable? Written mathematically,

$$(\exists\, x \in \mathbb{Z})(\forall\, y \in \mathbb{Z})(x + y = 1) \tag{31}$$

Explain your answer.

Answer: The statement is `False`. It demands a fixed x that makes $x + y = 1$ for every y. But, $x = 1 - y$ is not fixed, it changes for every y. Notice the similarity of equations 30 and 31. Apparently, $\forall$ and $\exists$ do not commute.

4 Now, let's restrict x and y be natural numbers (x, $y \in \mathbb{N}$).

4.1 Is the statement: "There is an x such that $x + y = 1$" `True` or `False` or undecidable? Written mathematically,

$$(\exists\, x \in \mathbb{N})(x + y = 1) \tag{32}$$

Answer: The statement is only `True` for $y = 0$ and $y = 1$. In these case, $x = 1 - y$ is also a natural number. The statement is `False` for all other values of y. Without knowing the value of y the statement is undecidable.

4.2 Is the statement: "There is an x such that for every y, $x + y = 1$" `True` or `False` or undecidable? Written mathematically,

$$(\exists\, x \in \mathbb{N})(\forall\, y \in \mathbb{N})(x + y = 1) \tag{33}$$

Explain your answer.

Answer: The statement is `False`. There is no single fixed natural number x such that $x + y = 1$ for every natural number y.

Back to the quiz on page 311.
Back to the notes on page 84.

Summative exam #1 (Key)

1 (10 pts) Fill in the truth table for De Morgan's first law: If P and Q are not both `True`, then one of P or Q is `False`. Conversely, if one of P or Q is `False`, then not both of them are `True`.

$$\neg(P \wedge Q) \equiv \neg P \vee \neg Q$$

De Morgan's First Law

Input		Computations	
P	Q	$\neg(P \wedge Q)$	$\neg P \vee \neg Q$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

2 (10 pts) Construct a truth table for the Boolean expression.

$$[(P \rightarrow Q) \wedge (\neg P \rightarrow R)] \rightarrow (Q \vee R)$$

Answer:

Input					Output		
P	Q	R	$(P \rightarrow Q)$	$\wedge$	$(\neg P \rightarrow R)$	$\rightarrow$	$(Q \vee R)$
0	0	0	1	0	0	1	0
0	0	1	1	1	1	1	1
0	1	0	1	0	0	1	1
0	1	1	1	1	1	1	1
1	0	0	0	0	1	1	0
1	0	1	0	0	1	1	1
1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1

3 (10 pts) Given the truth table below, find a Boolean expression that computes it.

	Input			Output
Row	A	B	C	D
1	0	0	0	0
2	0	0	1	0
3	0	1	0	0
4	0	1	1	1
5	1	0	0	0
6	1	0	1	1
7	1	1	0	1
8	1	1	1	1

Answer: Taking the row where the output D is 1. Form an AND-clauses of the input to make the row `True`. Form the OR of these clauses.

$$D = (\neg A \wedge B \wedge C) \vee (A \wedge \neg B \wedge C) \vee (A \wedge B \wedge \neg C) \vee (A \wedge B \wedge C)$$

Alternatively, take the row where the output D is 0. Form an OR-clauses of the input to make the row `False`. Form the AND of these clauses.

$$D = (A \vee B \vee C) \wedge (A \vee B \vee \neg C) \wedge (A \vee \neg B \vee C) \wedge (\neg A \vee B \vee C)$$

4 (10 pts) The Internet Protocol (IP) is used to route traffic on the Internet. IPv6 uses 128-bits (16-bytes) to name addresses. What is the size of Internet version 6 space?

Answer: Internet version 6 space has $2^{128} = 16^{32}$ addresses. A simple approximation gives

$$\begin{aligned} 2^{128} &= (2^{10})^{12.8} \\ &\approx (10^3)^{12.8} \\ &= 10^{38.4} \end{aligned}$$

SI (International System of Units) prefixes go up to *yotta*: 10^{24}. So there are about one hecto-tera-yotta-addresses (10^{38}) in IPv6 space.

5 (10 pts) Using n digits what is the largest natural number that can be written? Using n bits what is the largest natural number that can be written?

Answer: The largest decimal number that can written using n digits is $10^n - 1$. The largest binary number that can written using n bits is $2^n - 1$.

6 (10 pts) How many bits does it take to write the number 73? How many bits does it take to write the number n?

Answer: 73 can be written in 7 bits. Note that repeated division (quotients and remainders) is a simple way to compute a number's binary representation. For instance,

Repeated Remaindering mod 2

QUOTIENTS	73	36	18	9	4	2	1
REMAINDERS	1	0	0	1	0	0	1

Therefore $73 = (1001001)_2$.

In general, any number $n \geq 1$ can be bound below and above by powers of 2

$$2^k \leq n < 2^{k+1} \quad \text{for some } k \geq 0.$$

In this case, it will require $k+1$ bits to write n. Using the inequalities above, and taking the log base 2 of each term, deduce that

$$k \leq \lg n < k + 1$$

Take the floor of $\lg n$ to get k and add 1 to get $k+1$, the number of bits needed to write n.

$$\lfloor \lg n \rfloor + 1 = \text{number of bits to write } n.$$

7 (30 pts) Let $n, m \in \mathbb{N}$ be natural numbers. Write the following statements using mathematical notation.

7.1 There is an $n \in \mathbb{N}$ such that $m = n/2$.
Answer:

$$(\exists n \in \mathbb{N})(m = n/2)$$

7.2 For all $m \in \mathbb{N}$, there is an $n \in \mathbb{N}$ such that $n/2 = m$.
Answer:

$$(\forall m \in \mathbb{N})(\exists n \in \mathbb{N})(m = n/2)$$

7.3 There is an $n \in \mathbb{N}$ such that for all $m \in \mathbb{N}$, $m = n/2$.
Answer:

$$(\exists n \in \mathbb{N})(\forall m \in \mathbb{N})(m = n/2)$$

Which of these three statements are `True`, `False`, or `undecidable`.

Answer:

7.1 Question (7.1) is `True`: $n = 2m$ is a natural number if m is and $m = n/2$.

7.2 Question (7.2) is `True`: Since there were no restrictions on m in the previous answer, the argument applies to all m.

7.3 Question (7.3) is `False`: It says there is some fixed natural number n such that every other natural number is twice n ($m = 2n$).

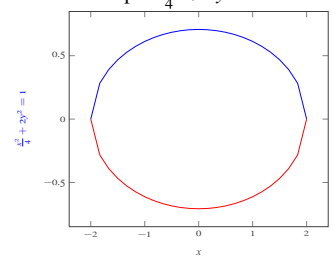
8 (10 pts) Okay, I have to know, who can solve my favorite quadratic equation?

$$x^2 - x - 1 = 0$$

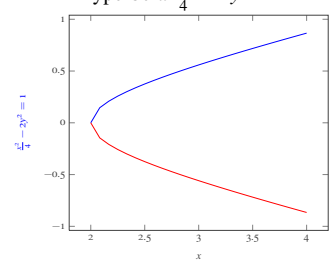
The roots are interesting numbers, for 10 extra points, tell me about the roots.

Back to the quiz on page 312.
Back to the notes on page 88.

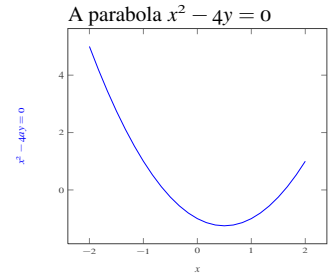
An ellipse $\frac{x^2}{4} + 2y^2 = 1$



An hyperbola $\frac{x^2}{4} - 2y^2 = 1$



A parabola $x^2 - 4y = 0$



Keys to Quiz(zes) on Functions

Function Basics (Key)

Answer:

1 Write, in English, an explanation of what it means to say: $f : \mathbb{X} \to \mathbb{Y}$ is a function from $\mathbb{X}$ to $\mathbb{Y}$.

Answer: In English you could say:

- "f maps each x in $\mathbb{X}$ to one and only one y in $\mathbb{Y}$."
- "f describes how each x in $\mathbb{X}$ is assigned to one and only one y in $\mathbb{Y}$."
- "f assigns, by some rule, each element x in the *domain* $\mathbb{X}$ to a single value y *co-domain* $\mathbb{Y}$."

This is the concept of a *total* function. There is also the idea of a *partial* function, where some elements x in $\mathbb{X}$ may not be mapped to an element $y \in \mathbb{Y}$.

2 Use predicate logic to write what it means to say: $f : \mathbb{X} \to \mathbb{Y}$ is a function from $\mathbb{X}$ to $\mathbb{Y}$.

Answer: It is simple to say $f(x) = y$. The trick is to restrict y to only one value. You can do this by saying: If x mapped to two (or more values) of y, then the y's are in fact equal.

$$(\forall x \in \mathbb{X})(\forall y_0, y_1 \in \mathbb{Y})((f(x) = y_0) \wedge (f(x) = y_1) \Rightarrow (y_0 = y_1))$$

This is the concept of a *total* function. There is also the concept of a *partial* function. Such a function may not be defined on every element in the *domain* $\mathbb{X}$.

$$(\exists x \in \mathbb{X})(\forall y_0, y_1 \in \mathbb{Y})((f(x) = y_0) \wedge (f(x) = y_1) \Rightarrow (y_0 = y_1))$$

3 Write, in English and in predicate logic, statements that describe what it means to say: $f : \mathbb{X} \to \mathbb{Y}$ is an onto function from $\mathbb{X}$ to $\mathbb{Y}$.

Answer: In English, some elocutions are:

- "f is a function and for each y in $\mathbb{Y}$ to an x in $\mathbb{X}$ such that $f(x) = y$."
- "f is a function and every y in $\mathbb{Y}$ is mapped by f to by some x in $\mathbb{X}$."
- "f is a function and the range of f is the co-domain $\mathbb{Y}$."

In predicate logic you could write

$$(\forall y \in \mathbb{Y})(\exists x \in \mathbb{X})(f(x) = y)$$

This assumes, by context, that f is known to be a function.

4 Write explanations in English and predicate logic of what it means to say $f : \mathbb{X} \to \mathbb{Y}$ is a one-to-one function.

Answer: In English: f is a function and each y in $\mathbb{Y}$ is the image of at most one x in $\mathbb{X}$. Or, different input map to different output.

$$(\forall u,\, v \in \mathbb{X})((u \neq v) \Rightarrow (f(u) \neq f(v)))$$

5 Do you know your conic sections? Which conics define functions from $\mathbb{X}$ to $\mathbb{Y}$?

5.1 The parabola: $ax^2 + y = 0$

Answer: This parabolic curve is a function from $\mathbb{X}$ to $\mathbb{Y}$. Each x maps to one and only one $y = -ax^2$.

5.2 The ellipse: $\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = 1$

Answer: This elliptical curve is not a function from $\mathbb{X}$ to $\mathbb{Y}$. For instance, when $x = 0$, the value of y can be b or $-b$. A function from $\mathbb{X}$ to $\mathbb{Y}$ maps each x to one and only one y.

5.3 The hyperbola: $\left(\frac{x}{a}\right)^2 - \left(\frac{y}{b}\right)^2 = 1$

Answer: This hyperbolic curve is not a function from $\mathbb{X}$ to $\mathbb{Y}$. For instance, when $x = \sqrt{2}a$, the value of y can be b or $-b$. A function from $\mathbb{X}$ to $\mathbb{Y}$ maps each x to one and only one y.

Back to the quiz on page 314.
Back to the notes on page 92.

The Pigeonhole Principle (Key)

Write $|\mathbb{X}| = 13$ if $\mathbb{X}$ has 13 members and say $\mathbb{X}$ has cardinality 13.

1 Suppose there are twelve people who must do 25 jobs.

1.1 Is it possible to define an onto function that maps people to jobs? If so, explain how.

Answer: It is *not* possible to define an onto function. Each person would have to do two or more jobs and that is not a function. A function from people to jobs would leave some jobs undone.

1.2 Is it possible to define an onto function from jobs to people? If so, explain how.

Answer: It is possible to define an onto function from jobs to people. Some people will have more than one job. Someone would need to have at least 3 jobs: If everyone did 2 or fewer jobs, at most 24 jobs would be done.

1.3 Is it possible to define a one-to-one function from people to jobs? If so, explain how.

Answer: Well, yes it is possible, but not all jobs would be covered.

1.4 Is it possible to define a one-to-one function from jobs to people? If so, explain how.

Answer: No it is *not* possible. Some person would have to be assigned more than one job.

2 A bridge hand is 13 cards from a standard deck of 52 cards.

2.1 Why must there be at least 4 cards of some suit in a bridge hand?

Answer: If there were only 3 or less cards of every suit, then there would be no more than 12 cards in the hand.

2.2 Why is there some suit with only 3 cards in a bridge hand?

Answer: If there were 4 or more cards of every suit, then there would be more than 16 cards in the hand.

Back to the quiz on page 314.
Back to the notes on page 98.

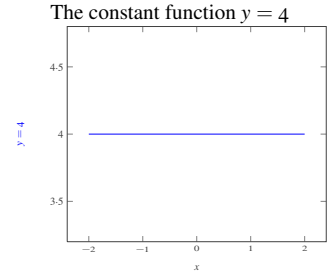
Examples Functions (Key)

1 Draw a graph of the following polynomial functions.

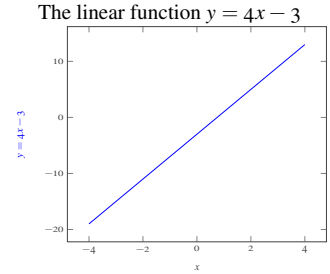
Answer:

- $y = 4$
- $y = 4x - 3$
- $y = x^2 - x - 1$

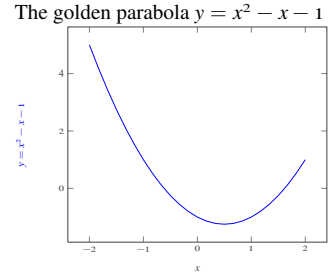
The constant function $y = 4$



The linear function $y = 4x - 3$



The golden parabola $y = x^2 - x - 1$



2 Compute the following values.

2.1 $\lg(\sqrt[3]{16})$

Answer: $\lg(\sqrt[3]{16}) = 4/3$ because $2^{4/3} = \sqrt[3]{16}$.

2.2 $\lg(\sqrt[5]{8})$

Answer: $\lg(\sqrt[5]{8}) = 3/5$ because $2^{3/5} = \sqrt[5]{8}$.

2.3 $\lg \sqrt[3]{32}$

Answer: The logarithm base 2 of the cube root of 32 is 5/3 because $2^{5/3} = \sqrt[3]{32}$.

2.4 $\log_{16} \sqrt[3]{32}$

Answer: The logarithm base 16 of the cube root of 32 is 5/12. Note that

$$\log_{16} \sqrt[3]{32} = \frac{1}{3} \log_{16} 32 = \frac{1}{3} \cdot \frac{5}{4} = \frac{5}{12}$$

2.5 $\lg(2^{n!})$

Answer: $\lg(2^{n!}) = n!$

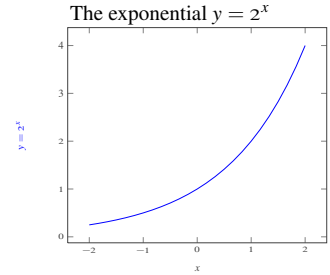
Answer:

3 Draw a graph of the following logarithms or exponential functions.

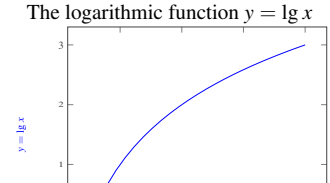
3.1 $y = 2^x$

3.2 $y = \lg x$

The exponential $y = 2^x$



The logarithmic function $y = \lg x$



4 Identify which of the following is a permutation of the digits

$$\mathbb{D} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

For those that are, describe them using cycle notation.

4.1 $\langle 3, 4, 5, 9, 8, 7 \rangle$
Answer: This is not a permutation of the digits because it is not onto $\mathbb{D}$.

4.2 $\langle 5, 4, 3, 2, 1, 0, 6, 7, 9, 8 \rangle$
Answer: This is a permutation of the digits. It is a one-to-one and onto function. In cycle notation it is

$$[1, 4][2, 3][5, 0][6][7][8, 9]$$

4.3 $\langle 5, 5, 4, 3, 1, 1, 0, 6, 7, 8, 9 \rangle$
Answer: This is not a permutation because 1 is repeated twice.

Back to the quiz on page 314.
Back to the notes on page 107.

Keys to Quiz(zes) on Horner's Rule

Horner's rule (Key)

1 Use Horner's rule to evaluate the polynomials at the indicated value of x. How many multiplications and additions were required in each case?

1.1 $p(x) = -7x^2 + 28x + 7$ at $x = 3$.
Answer:

Horner's Rule @ $x = 3$

-7	28	7
	-21	21
-7	7	$\boxed{28}$

Therefore $p(3) = 28$. The computation used 2 multiplies and 2 additions.

1.2 $p(x) = 4x^4 + 5x^2 - 3x + 2$ at $x = 2$.
Answer:

Horner's Rule @ $x = 2$

4	0	5	-3	2
	8	16	42	78
4	8	21	39	$\boxed{80}$

Therefore $p(2) = 80$. The computation used 4 multiplies and 4 additions.

1.3 $p(x) = 3x^5 - x^3 + 4x^2 + x + 1$ at $x = -2$.

Answer:

Horner's Rule @ $x = -2$

3	0	−1	4	1	1
	−6	12	−22	36	−74
3	−6	11	−18	37	$\boxed{-73}$

Therefore $p(-2) = -73$. The computation used 5 multiplies and 5 additions.

2 How many multiplies and additions does Horner's rule make when evaluating a polynomial of degree $n - 1$?

Answer: The computation will require $n - 1$ multiplies and $n - 1$ additions.

Back to the quiz on page 315.
Back to the notes on page 118.

Horner's rule for conversion to decimal (Key)

Use Horner's rule to convert the following natural numbers (unsigned integers) into decimal notation.

1 $(0000\ 1001)_2$

Answer:

Horner's Rule @ $x = 2$

0	0	0	0	1	0	0	1
	0	0	0	0	2	4	8
0	0	0	0	1	2	4	$\boxed{9}$

$\therefore (0000\ 1001)_2 = (9)_{10}$.

2 $(1111\ 1111)_2$

Answer:

Horner's Rule @ $x = 2$

1	1	1	1	1	1	1	1
	2	6	14	30	62	126	254
1	3	7	15	31	63	127	$\boxed{255}$

$\therefore (1111\ 1111)_2 = (255)_{10}$.

3 $(0100\ 1100)_2$

Answer:

Horner's Rule @ $x = 2$

0	1	0	0	1	1	0	0
	0	2	4	8	18	38	76
0	1	2	4	9	19	38	$\boxed{76}$

$\therefore (0100\ 1100)_2 = (76)_{10}$.

4 $(747)_8$

Answer:

Horner's Rule @ $x = 8$

7	4	7
	56	480
7	60	$\boxed{487}$

$\therefore (747)_8 = (487)_{10}$.

5 $(123)_4$

Answer:

Horner's Rule @ $x = 4$

1	2	3
	4	24
1	6	$\boxed{27}$

$\therefore (123)_4 = (27)_{10}$.

6 $(BAD)_{16}$

Answer:

Horner's Rule @ $x = 16$

11	10	13
	176	2976
11	186	$\boxed{2989}$

$\therefore (BAD)_{16} = (2989)_{10}$.

Back to the quiz on page 315.
Back to the notes on page 121.

Remaindering to convert from decimal to another base (Key)

1 Use repeated remaindering to convert the following decimal numbers to their representation in the indicated base.

1.1 Convert 76 to binary:

Answer: Quotients and remainders in the table below are computed from right-to-left.

Repeated Remaindering mod 2

QUOTIENTS	76	38	19	9	4	2	1
REMAINDERS	0	0	1	1	0	0	1

$\therefore 76 = (100\ 1100)_2$

1.2 Convert 137 to binary:

Answer:

Repeated Remaindering mod 2

QUOTIENTS	137	68	34	17	8	4	2	1
REMAINDERS	1	0	0	1	0	0	0	1

$\therefore 137 = (1000\ 1001)_2$

1.3 Convert 177 to octal:

Answer:

Repeated Remaindering mod 8

QUOTIENTS	177	22	2
REMAINDERS	1	6	2

$\therefore 177 = (261)_8$

2 Conversion between binary, octal, and hexadecimal can be done by grouping or ungrouping bits.

2.1 Group the bits you computed in question (1.1) and (1.2) three at a time to write 76 and 137 in octal.

Answer: For 76 write

$$(1\ 001\ 100)_2 = (114)_8$$

For 137 write

$$(10\ 001\ 001)_2 = (211)_8$$

2.2 Group the bits you computed in question (1.1) and (1.2) four at a time to write 76 and 137 in hexadecimal.

Answer: For 76 write

$$(100\ 1100)_2 = (4C)_{16}$$

For 137 write

$$(1000\ 1001)_2 = (89)_{16}$$

Back to the quiz on page 316.
Back to the notes on page 124.

Keys to Quiz(zes) on Sequences

Sequence basics (Key)

1 What is the sum of the first n terms of the Alice sequence? How are the sequence of these sums related to the Gauss sequence?

Answer: The sum of n 1's is n. The sequence of sums of n terms from the Alice sequence is the Gauss sequence.

$$\begin{aligned}
= 0 &\quad \text{sum of no } (n=0) \text{ term} \\
1 = 1 &\quad \text{sum of one } (n=1) \text{ term} \\
1+1 = 2 &\quad \text{sum of two } (n=2) \text{ terms} \\
1+1+1 = 3 &\quad \text{sum of three } (n=3) \text{ terms} \\
1+1+1+\cdots+1 = n &\quad \text{where there are } n \text{ terms in the sum.}
\end{aligned}$$

2 What is the sum of the first n terms of the Gauss sequence? How are the sequence of these sums related to the Triangular sequence?

Answer: The sum $0+1+2+\cdots+(n-1)$ is $n(n-1)/2$. The sequence of sums of n terms from the Alice sequence is the Gauss sequence.

$$\begin{aligned}
0 &= \frac{0(0-1)}{2} && \text{sum of no } (n=0) \text{ terms} \\
0 = 0 &= \frac{1(1-1)}{2} && \text{sum of one } (n=1) \text{ term} \\
0+1 = 1 &= \frac{2(2-1)}{2} && \text{sum of two } (n=2) \text{ terms} \\
0+1+2 = 3 &= \frac{3(3-1)}{2} && \text{sum of three } (n=3) \text{ terms} \\
0+1+2+3 = 6 &= \frac{4(4-1)}{2} && \text{sum of three } (n=4) \text{ terms} \\
0+1+2+\cdots+(n-1) = & \quad \frac{n(n-1)}{2} && \text{sum of } n \text{ terms}
\end{aligned}$$

3 Consider the sequence of even natural numbers

$$\langle 0, 2, 4, 6, 8, \ldots \rangle$$

3.1 What function $e(n)$ maps the natural numbers to the even natural numbers?

Answer: The function is $e(n) = 2n$.

3.2 What is the recursive-initial value form for the even natural numbers?

Answer: The recursion and initial condition is

$$e_n = e_{n-1} + 2, \qquad e_0 = 0$$

4 Consider the sequence of natural numbers that have a remainder of 1 when divided by 3

$$\langle 1, 4, 7, 10, 13, \ldots \rangle$$

4.1 What function maps the natural numbers to these numbers?

Answer: The function is $f(n) = 3n + 1$.

4.2 What is the recursive-initial value form for these numbers?

Answer: The recursion and initial condition is

$$f_n = f_{n-1} + 3, \qquad f_0 = 1$$

5 The *half-life* sequence is the decreasing sequence

$$\left\langle 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \ldots \right\rangle$$

5.1 What function maps the natural numbers to these numbers?

Answer: The function is $d(n) = (1/2)^n$.

5.2 What is the recursive-initial value form for these numbers?

Answer: The recursion and initial condition is

$$d_n = \frac{d_{n-1}}{2}, \qquad d_0 = 1$$

Back to the quiz on page 316.
Back to the notes on page 132.

Differences (Key)

1 Given the sequences below, what is the function for a term in the sequence and what function computes the difference of the sequence?

1.1 The *quadratic* sequence

$$\vec{R} = \langle 0,\ 1,\ 4,\ 9,\ 16, \ldots \rangle$$

Answer: The general term is given by $q(n) = n^2$. The function for the difference is

$$\triangle n^2 = (n+1)^2 - n^2 = 2n + 1$$

1.2 The *cubic* sequence

$$\vec{R} = \langle 0,\ 1,\ 8,\ 27,\ 64, \ldots \rangle$$

Answer: The general term is given by $c(n) = n^3$. The function for the difference is

$$\triangle n^3 = (n+1)^3 - n^3 = 3n^2 + 3n + 1$$

1.3 The *ratio* sequence

$$\vec{R} = \langle 1,\ r,\ r^2,\ r^3,\ r^4, \ldots \rangle$$

Answer: The general term is given by $r(n) = r^n$. The function for the difference is

$$\triangle r^n = r^{n+1} - r^n = r^n(r-1)$$

1.4 The *logarithm* sequence

$$\vec{L} = \langle \lg 1,\ \lg 2,\ \lg 3,\ \lg 4,\ \lg 5, \ldots \rangle$$

Answer: The general term is given by $l(n) = \lg n$. The function for the difference is

$$\triangle \lg n = \lg n + 1 - \lg n = \lg\left(\frac{n+1}{n}\right)$$

You will not be held responsible for trigonometric functions, but you may need to use them in other classes.

1.5 The *sine* sequence

$$\vec{S} = \langle \sin 0,\ \sin 1,\ \sin 2,\ \sin 3,\ \sin 4, \ldots \rangle$$

Answer: The general term is given by $s(n) = \sin n$. The function for the difference is

$$\triangle \sin n = \sin(n+1) - \sin n = 2\sin\left(\frac{1}{2}\right)\cos\left(n + \frac{1}{2}\right)$$

Back to the quiz on page 317.
Back to the notes on page 133.

Sums (Key)

1 Consider the general sum

$$\sum_{k=0}^{n-1} t_k$$

1.1 What is the lower limit?

Answer: The lower limit on the sum is $k = 0$.

1.2 What is the upper limit?

Answer: The upper limit on the sum is $k = n - 1$.

1.3 What terms are being summed?

Answer: The terms in the sum are $t_0, t_1, t_2, \ldots, t_{n-1}$.

2 What are the functions that compute the following sums?

2.1 The sum of terms in the Alice sequence

$$\sum_{k=0}^{n-1} 1$$

Note that $1 = (k+1) - k$.

Answer: The sum is equal to n. These are the terms in the Gauss sequence: $\langle 0, 1, 2, 3, \ldots, n, \ldots \rangle$.

2.2 The sum of terms in the Gauss sequence

$$\sum_{k=0}^{n-1} k$$

Note that $k = [(k+1)k - k(k-1)]/2$.

Answer: The sum is equal to $n(n-1)/2$. These are the terms in the Triangular sequence: $\langle 0, 0, 1, 3, \ldots, n(n-1)/2, \ldots \rangle$.

2.3 The sum of terms in the Doubling sequence

$$\sum_{k=0}^{n-1} 2^k$$

Note that $2^k = 2^{k+1} - 2^k$.

Answer: The sum is equal to $n(n-1)/2$. These are the terms in the Triangular sequence: $\langle 0, 0, 1, 3, \ldots, n(n-1)/2, \ldots \rangle$.

Back to the quiz on page 317.
Back to the notes on page 135.

Fundamental Theorem (Key)

1 Use the fact that

$$\triangle x^{\underline{m+1}} = (m+1)x^{\underline{m}}$$

Recall from the notes on Horner's rule falling factorial powers are defined by

$$x^{\underline{m}} = x(x-1)(x-2)\cdots(x-m+1)$$

and it was shown that

$$\triangle x^{\underline{m}} = m x^{\underline{m-1}}$$

and the fundamental theorem of the sum & difference calculus to find a simple formula for

$$\sum_{k=0}^{n-1} k^{\underline{m}}$$

Answer: You have

$$\begin{aligned}\sum_{k=0}^{n-1} k^{\underline{m}} &= \sum_{k=0}^{n-1} \frac{\triangle k^{\underline{m+1}}}{m+1} \\ &= \left.\frac{k^{\underline{m+1}}}{m+1}\right|_0^n \\ &= \frac{n^{\underline{n+1}}}{m+1}\end{aligned}$$

Sums of *ordinary* powers are more complex.

$$\sum_{k=0}^{n-1} k^0 = n$$

$$\sum_{k=0}^{n-1} k^1 = \frac{1}{2}n^2 - \frac{1}{2}n$$

$$\sum_{k=0}^{n-1} k^2 = \frac{1}{3}(n-1)^3 - \frac{2}{2}(n-1)^2 + \frac{1}{6}(n-1)$$

and lead to the study of Bernoulli numbers.

2 What is a simple formula for the sum

$$\sum_{k=1}^{n} \lg\left(\frac{k+1}{k}\right)$$

Answer: Since the log of a quotient is the difference of logs you can write

$$\sum_{k=1}^{n} \lg\left(\frac{k+1}{k}\right) = \sum_{k=1}^{n} (\lg(k+1) - \lg k)$$

which is equal to

$$(\lg 2 - \lg 1) + (\lg 3 - \lg 2) + (\lg 4 - \lg 3) + \cdots + (\lg n - \lg(n-1)) + (\lg(n+1) - \lg n)$$

and this sum telescopes to

$$\lg(n+1) - \lg 1 = \lg(n+1)$$

Back to the quiz on page 318.
Back to the notes on page 136.

Summative exam #2 (Key)

1 (10 pts) Write an English statement to explain that $f : \mathbb{X} \to \mathbb{Y}$ is an onto function. Using the notation of predicate logic, write the definition that $f : \mathbb{X} \to \mathbb{Y}$ is an onto function from $\mathbb{X}$ to $\mathbb{Y}$.

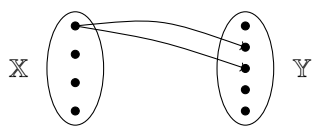
Answer: f is a function and for each y in $\mathbb{Y}$ there is an x in $\mathbb{X}$ that f maps to y.

$$(\forall y \in \mathbb{Y})(\exists x \in \mathbb{X})(f(x) = y)$$

2 (20 pts) By drawing arrows from points in $\mathbb{X}$ to points in $\mathbb{Y}$ show how to construct examples of the following types, or explain why no such example can be drawn.

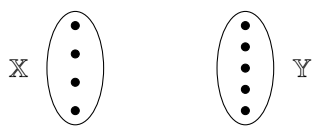
2.1 Draw a picture of a graph that is not a function.

Answer: Here is one solution: The top x value maps to two different y's.



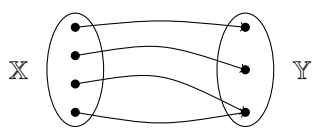
2.2 Draw a picture of a function that is onto.

Answer: You cannot draw an onto function in this case. Since there are 5 y's and only 4 x's, some y will not be the image of any x.



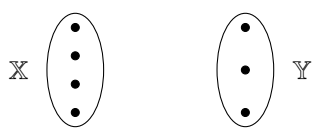
2.3 Draw a picture of a function that is onto.

Answer: Here is one solution:



2.4 Draw a picture of a function that is one-to-one.

Answer: You cannot draw a one-to-one function in this case. Each x maps to some y. Since there are 4 x's and only 3 y's, some y will have 2 or more x's that map to it.



3 (10 pts) Use Horner's rule to evaluate the polynomial

$$p(x) = 3x^5 - 20x^3 - 60x - 7$$

and $x = 3$.

Answer:

Horner's Rule @ $x = 3$

3	0	−20	0	−60	−7
	9	27	21	63	9
3	9	7	21	3	$\boxed{2}$

$\therefore\ p(3) = 2$

4 (10 pts) Use Horner's rule to convert the binary number $(1110\ 1101)_2$ to decimal notation.

Answer:

Horner's Rule @ $x = 2$

1	1	1	0	1	1	0	1
	2	6	14	28	58	118	236
1	3	7	14	29	59	118	$\boxed{237}$

$\therefore\ (1110\ 1101)_2 = (237)_{10}$

5 (10 pts) Use repeated remaindering to convert the decimal number $(87)_{10}$ to binary notation.

Answer:

Repeated Remaindering mod 2

QUOTIENTS	87	43	21	10	5	2	1
REMAINDERS	1	1	1	0	1	0	1

$\therefore\ (87)_{10} = (1010\ 111)_2$

6 (10 pts) What decimal numbers do the hexadecimal numbers A, B, C, D, E, and F represent.

Answer:

$$A = 10, \quad B = 11, \quad C = 12, \quad D = 13, \quad E = 14, \quad F = 15$$

7 (10 pts) Part (a) is simple arithmetic to help you guess the general answer in part (b).

7.1 Verify that the sum on the left is equal to the expression on the right.

7.1.1 $1+2=\frac{2\cdot 3}{2}$
Answer: $1+2=3=\frac{6}{2}=\frac{2\cdot 3}{2}$

7.1.2 $1+2+3=\frac{3\cdot 4}{2}$
Answer: $1+2+3=6=\frac{12}{2}=\frac{3\cdot 4}{2}$

7.1.3 $1+2+3+4=\frac{4\cdot 5}{2}$
Answer: $1+2+3+4=10=\frac{20}{2}=\frac{4\cdot 5}{2}$

7.2 Compute the value of the Gauss sum

$$1+2+3+4+5+\cdots+98+99+100$$

Answer:

$$1+2+3+4+5+\cdots+98+99+100=\frac{100\cdot 101}{2}=5050$$

8 (10 pts) Part (a) is simple arithmetic to help you guess the general answer in part (b).

8.1 Verify that the sum on the left is equal to the expression on the right.

8.1.1 $1+2=2^2-1$
Answer: $1+2=3=4-1=2^2-1$

8.1.2 $1+2+4=2^3-1$
Answer: $1+2+4=7=8-1=2^3-1$

8.1.3 $1+2+4+8=2^4-1$
Answer: $1+2+4+8=15=2^4-1$

8.2 Compute the value of the Mersenne sum (you are expected to leave your answer in exponential form).

$$1+2+4+8+16+\cdots+2^{98}+2^{99}+2^{100}$$

Answer:

$$1+2+4+8+\cdots+2^{98}+2^{99}+2^{100}=2^{101}-1$$

9 (10 pts) I once gave a 20 question `True`/`False` exam.

9.1 In how many ways can you answer the questions (pretend you answer each question `True` or `False`)?
Answer: There are 2^{20} ways: There are two answers (`True` or `False`) that can be chosen for each question.

9.2 If you decide to leave some questions blank, in how many ways can you answer the questions?
Answer: There are 3^{20} ways: There are three answers (`True`, `False`, or blank) that can be chosen for each question.

Back to the quiz on page 319.
Back to the notes on page 139.

Keys to Quiz(zes) on Machine Numbers

Machine Number Basics (Key)

1 What is the largest unsigned natural number on:

1.1 A 16-bit computer?
Answer: The largest 16 bit number is the Mersenne number $2^{16} - 1 = 65,535$, about 65 thousand.

1.2 A 32-bit computer?
Answer: The largest 32 bit number is the Mersenne number $2^{32} - 1 = 4,294,967,295$, about 4 billion.

1.3 A 64-bit computer?
Answer: The largest 64 bit number is the Mersenne number $2^{64} - 1 = 18,446,744,073,709,551,615$, about 18 quintillion.

2 What is the decimal value of the following ten's complement numbers? Assume the word length is 4.

2.1 $(3084)_{10c}$
Answer: $(3084)_{10c} = 3084$.

2.2 $(8034)_{10c}$
Answer: $(8034)_{10c} = 8034 - 10000 = 1966$.

3 Pad the following ten's complement numbers to be 5 digits long.

3.1 $(384)_{10c}$
Answer: $(384)_{10c} = (00384)_{10c}$.

3.2 $(834)_{10c}$
Answer: $(834)_{10c} = (99834)_{10c}$.

Back to the quiz on page 321.
Back to the notes on page 143.

Two's Complement Notation (Key)

1 Using two's complement notation what range of integers from most negative to most positive can be represented using

1.1 2 bits?
Answer: The range is from $-2^1 = -2$ to $2^1 - 1 = 1$.

$$(00)_{2c} = 0, \quad (01)_{2c} = 1, \quad (10)_{2c} = -2, \quad (11)_{2c} = -1$$

1.2 8 bits?
Answer: The range is from $-2^7 = -128$ to $2^7 - 1 = 127$.

$$(000)_{2c} = 0, \quad (001)_{2c} = 1, \quad (010)_{2c} = 2, \quad (011)_{2c} = 3$$
$$(100)_{2c} = -4, \quad (101)_{2c} = -3, \quad (110)_{2c} = -2, \quad (111)_{2c} = -1$$

1.3 32 bits?

Answer: The range is from $-2^{31} = -2,147,483,648$ to $2^{31} - 1 = 2,147,483,647$.

1.4 64 bits?

Answer: The range is from -2^{63} to $2^{63} - 1$. How big is 2^{63}? Approximate it by a power of 10.

2 Negate the two's complement integers below.

2.1 $(0100\ 1100)_{2c}$.

Answer: The negative of $(0100\ 1100)_{2c}$ is $(1011\ 0100)_{2c}$. The bit-flipping rule is: Copy the bits from right-to-left up to and including the first 1. Flip the remaining bits on the left. Notice the two numbers sum to 1 0000 0000 which is truncated to 8 bits all of them are 0.

2.2 $(1100\ 0000)_{2c}$.

Answer: The negative of $(1100\ 0000)_{2c}$ is $(0100\ 0000)_{2c}$.

2.3 $(1010\ 0100)_{2c}$.

Answer: The negative of $(1010\ 0100)_{2c}$ is $(0101\ 1100)_{2c}$.

2.4 $(1000\ 0000)_{2c}$.

Answer: You might say this is a "trick" question. Extend the sign by writing $(1000\ 0000)_{2c} = (1\ 1000\ 0000)_{2c}$ and then negate this last string to get $(0\ 1000\ 0000)_{2c}$.

3 Convert the following two's complement integers into decimal notation.

3.1 $(0100\ 1100)_{2c}$.

Answer: Conversion algorithms are discussed starting on page 145. The leftmost sign bit, 0, declares the number is positive. Use Horner's rule to compute

Horner's Rule @ $x = 2$

0	1	0	0	1	1	0	0
	0	2	4	8	18	38	76
0	1	2	4	9	19	38	$\boxed{76}$

Since the two's complement number is positive, $+76$ is the answer.

3.2 $(1011\ 0100)_{2c}$.

Answer: The leftmost sign bit, 1, declares the number is negative. Use Horner's rule to compute

Horner's Rule @ $x = 2$

1	0	1	1	0	1	0	0
	2	4	10	22	44	90	180
1	2	5	11	22	45	90	$\boxed{180}$

Since the two's complement number is negative the answer is $180 - 256 = -76$.

3.3 $(1010\ 1010)_{2c}$.

Answer: The leftmost sign bit, 1, declares the number is negative. Use Horner's rule to compute

Horner's Rule @ $x = 2$

1	0	1	0	1	0	1	0
	2	4	10	20	42	84	170
1	2	5	10	21	42	85	$\boxed{170}$

Since the two's complement number is negative the answer is $170 - 256 = -86$.

3.4 $(0010\ 1000)_{2c}$.

Answer: The leftmost sign bit, 0, declares the number is positive. Use Horner's rule to compute

Horner's Rule @ $x = 2$

0	0	1	0	1	0	0	0
	0	0	2	4	10	20	40
0	0	1	2	5	10	20	$\boxed{40}$

Since the two's complement number is positive the answer is $+40$.

4 Convert the signed decimal integers below into two's complement notation.

Answer: Conversion algorithms are discussed starting on page 145.

4.1 $+76$.

Answer: First convert unsigned 76 to binary: Quotients and remainders in the table below are computed from left-to-right.

Repeated Remaindering mod 2

QUOTIENTS	76	38	19	9	4	2	1
REMAINDERS	0	0	1	1	0	0	1

Prefixing string 100 1100 with a sign bit gives the two's complement representation

$$+76 = (0100\ 1100)_{2c}$$

4.2 -76.

Answer: Using the result from problem 4.1 we know $+76 = (0100\ 1100)_{2c}$ and this can be negated by using the algorithm described on page 145. This gives the result

$$-76 = (1011\ 0100)_{2c}$$

4.3 -137.

Answer: First convert unsigned 137 to binary: Quotients and remainders in the table below are computed from left-to-right.

Repeated Remaindering mod 2

QUOTIENTS	137	68	34	17	8	4	2	1
REMAINDERS	1	0	0	1	0	0	0	1

Prefixing the bit string 1000 1001 with a sign bit gives the two's complement representation

$$+137 = (0\ 1000\ 1001)_{2c}$$

Negate $+137 = (0\ 1000\ 1001)_{2c}$ to get the answer

$$-137 = (1\ 0111\ 0111)_{2c}$$

4.4 $+177$.

Answer: First convert unsigned 177 to binary using repeated remaindering.

Repeated Remaindering mod 2

QUOTIENTS	177	88	44	22	11	5	2	1
REMAINDERS	1	0	0	0	1	1	0	1

Prefixing the bit string 1011 0001 with a sign bit gives the two's complement representation

$$+177 = (0\ 1011\ 1011)_{2c}$$

Back to the quiz on page 321.
Back to the notes on page 151.

Floating Point Notation (Key)

1 One rule for a number m written in biased notation is that m should be a natural number. What is the minimum value for a bias b to represent the range of integers shown below?

1.1 $-127 \leq n \leq 128$.

Answer: The minimum value for the bias b is 127. The give range is translated to $0 \leq m \leq 255$.

1.2 $-511 \leq n \leq 512$.

Answer: The minimum value for the bias b is 511. The given range is translated to $0 \leq m \leq 1023$.

1.3 $-49 \leq n \leq 50$.

Answer: The minimum value for the bias b is 49. The given range is translated to $0 \leq m \leq 99$.

1.4 $s \leq n \leq t$, where s and t are integers with $s \leq t$.

Answer: The minimum value for the bias b is $-s$. The given range is translated to $0 \leq m \leq t - s$.

2 One guideline for biased notation is numbers represented should be about half-and-half positive-and-negative. Given the ranges of biased numbers below, what values of the bias b satisfies this guideline?

2.1 $0 \leq n_{\text{bias}=?} \leq 255$.

Answer: When the bias b is 127, then integers from -127 to 128 can be represented. About half of them are negative $-127, -126, \ldots, -1$. About half of them are positive $1, 2, \ldots, 128$. When the bias b is 128, then integers from -128 to 127 can be represented. About half of them are negative $-128, -127, \ldots, -1$. About half of them are positive $1, 2, \ldots, 127$. In this course, the default is to choose the odd bias $b = 127$.

2.2 $0 \leq n_{\text{bias}=?} \leq 1023$.

Answer: When the bias b is 511, integers from -511 to 512 can be represented. When the bias b is 512, integers from -512 to 511 can be represented. In this course, the default is to choose the odd bias $b = 511$.

2.3 $0 \leq n_{\text{bias}=?} \leq 99$.

Answer: When the bias b is 49, integers from -49 to 50 can be represented. When the bias b is 50, integers from -50 to 49 can be represented. In this course, the default is to choose the odd bias $b = 49$.

2.4 $s \leq n_{\text{bias}=?} \leq t$, where s and t are integers with $s \leq t$.

Answer: Set the bias b to $\lfloor (t - s)/2 \rfloor$.

3 Convert the biased numbers below to decimal integers.

Answer: Conversion from biased to decimal notation is discussed on page 152.

3.1 $(6)_{\text{bias}=3}$.
Answer: Subrtract the bias 3 from 6 to get

$$(6)_{\text{bias}=3} = 6 - 3 = +3$$

3.2 $(76)_{\text{bias}=126}$.
Answer: Subrtract the bias 126 from 76 to get

$$(76)_{\text{bias}=126} = 76 - 126 = -50$$

3.3 $(137)_{\text{bias}=126}$.
Answer: Subrtract the bias 126 from 137 to get

$$(137)_{\text{bias}=126} = 137 - 126 = +11$$

3.4 $(177)_{\text{bias}=254}$.
Answer: Subrtract the bias 254 from 177 to get

$$(177)_{\text{bias}=126} = 177 - 254 = -77$$

4 The following binary strings are floating point numbers where the first (leftmost) bit is a sign bit, the next three bits are exponent bits written in biased notation with bias $b = 3$, and the last four bits are fraction bits. These floating point are normalized. What are decimal values of these floating point numbers?

4.1 $(1\ 001\ 1000)_{fp}$
Answer: This is a negative number (the sign bit is 1) with exponent $(001)_{b=3} = 1 - 3 = -2$ and fixed point value $(1.1000)_2 = 3/2$.

$$\therefore (1001\ 1000)_{fp} = -\frac{3}{2} \times 2^{-2} = -\frac{3}{8}$$

4.2 $(0101\ 1011)_{fp}$
Answer: This is a positive number (the sign bit is 0) with exponent $(101)_{b=3} = 5 - 3 = 2$ and fixed point value $(1.1011)_2 = 27/16$.

$$\therefore (0101\ 1011)_{fp} = \frac{27}{16} \times 2^{2} = -\frac{27}{4}$$

4.3 $(1111\ 1111)_{fp}$
Answer: This is a negative number (the sign bit is 1) with exponent $(111)_{b=3} = 7 - 3 = 4$ and fixed point value $(1.1111)_2 = 31/16$.

$$\therefore (1111\ 1111)_{fp} = -\frac{31}{16} \times 2^{4} = -31$$

4.4 $(0000\ 0001)_{fp}$

Answer: This is a positive number (the sign bit is 0) with exponent $(000)_{b=3} = 0 - 3 = -3$ and fixed point value $(1.0001)_2 = 17/16$.

$$\therefore (0000\ 0001)_{fp} = \frac{17}{16} \times 2^{-3} = -\frac{17}{128}$$

5 Using the 8 bit float point format, explain why $17/128$ is the smallest positive floating point number and why $16/128 = 1/8$ is not.

Answer: The smallest non-zero number is represented by the string $(0\ 000\ 0001)_{fp}$. This string is interpreted as

$$(-1)^0\,(1.0001)_2 \times 2^{0-3} = \frac{17}{16} \times \frac{1}{8} = \frac{17}{128}$$

On the other hand, $1/8 = (1.0000)_2 \times 2^{-3}$ and that would be written as the all zero string, which is reserved for the value 0.

Back to the quiz on page 322.
Back to the notes on page 156.

Keys to Quiz(zes) on Names

Naming Basics (Key)

1 Using the binary alphabet, how many things can you name with fixed-length strings of the given length?

1.1 1 bit?

Answer: 2 things can be named: 0 and 1.

1.2 2 bits?

Answer: 4 things can be named: 00, 01, 10, 11.

1.3 3 bits?

Answer: 8 things can be named:

000, 001, 010, 011, 100, 101, 110, 111

1.4 n bits?

Answer: 2^n things can be named.

2 Using the binary alphabet, how many things can you name with non-empty, variable-length strings?

2.1 1 or fewer bits?

Answer: 2 things can be named: 0 and 1.

2.2 2 or fewer bits?

Answer: 6 things can be named: 0, 1, 00, 01, 10, 11.

2.3 3 or fewer bits?

Answer: 14 things can be named:

$$0,\ 1,\ 00,\ 01,\ 10,\ 11,\ 000,\ 001,\ 010,\ 011,\ 100,\ 101,\ 110,\ 111$$

2.4 n or fewer bits?

Answer: The number of things that can be named using n or fewer bits is.

$$\sum_{k=1}^{n} 2^k = 2 + 4 + 8 + \cdots + 2^n = 2^{n+1} - 2$$

3 Using the English alphabet, how many things can you name with fixed-length strings of the given length?

3.1 1 character?

Answer: 26 things can be named:

$$a,\ b,\ c, \ldots,\ z$$

3.2 2 characters?

Answer: $26^2 = 676$ things can be named:

$$\{aa,\ ab,\ ac,\ \ldots,\ zx,\ zy,\ zz\}$$

3.3 3 characters?

Answer: $26^3 = 17,576$ things can be named:

$$\{aaa,\ aab,\ aac,\ \ldots,\ zzx,\ zzy,\ zzz\}$$

3.4 n characters?

Answer: 26^n things can be named.

4 Using the English alphabet, how many things can you name with non-empty, variable-length strings of the given length?

4.1 1 or fewer characters?

Answer: 26 things can be named:

$$a,\ b,\ c, \ldots,\ z$$

4.2 2 or fewer characters?

Answer: $26 + 26^2 = 702$ things can be named:

$$\{a,\ b,\ c, \ldots,\ z,\ aa,\ ab,\ ac,\ \ldots,\ zx,\ zy,\ zz\}$$

4.3 3 or fewer characters?

Answer: $26 + 26^2 + 26^3 = 18,278$ things can be named:

$$\{a, \ldots, z, aa, \ldots, zz, aaa, \ldots, zzz\}$$

4.4 n or fewwer characters?

Answer: The number of things that can be named using n or fewer characters is.

$$\sum_{k=1}^{n} 26^k = 26 + 26^2 + 26^3 + \cdots + 26^n = \frac{26^{n+1} - 26}{25}$$

Back to the quiz on page 323.
Back to the notes on page 167.

Naming Numbers (Key)

For instance, using the binary alphabet, you can name

- 1 or 2 things using 1 bit: (0) or (0 and 1).
- 3 or 4 things using 2 bits: (00, 01, 10) or (00, 01, 10, 11)
- 5 through 8 things using 3 bits: To name the things, choose 5 through 8 elements from the set

$$\{000, 001, 010, 011, 100, 101, 110, 111\}$$

Using the English alphabet, you can name

- 1 through 26 things using 1 letter: a through z.
- 27 through $26^2 = 676$ things using 2 letters To name the things, choose 27 through 676 elements from the set

$$\{aa, ab, ac, \ldots, zx, zy, zz\}$$

- 677 through $26^3 = 17,576$ things using 3 letters: To name the things, choose 677 through 17,576 elements from the set

$$\{aaa, aab, aac, \ldots, zzx, zzy, zzz\}$$

Back to the quiz on page 323.
Back to the notes on page 162.

Summative exam #3 (Key)

1 (20 pts) The sum $1 + 2 + 4 + \ldots + 2^{n-1}$ of powers of 2 numbers called *geometric*. What function (formula) computes the value of the sum

$$1 + 2 + 4 + \ldots + 2^{n-1} = \sum_{k=0}^{n-1} 2^k$$

Answer: The sum is a Mersenne number

$$1 + 2 + 4 + \ldots + 2^{n-1} = \sum_{k=0}^{n-1} 2^k = 2^n - 1$$

Not asked, but you should recognize that $1 + 2 + 4 + \ldots + 2^{n-1}$ is a binary number of length n where all the bits are 1.

2 (10 pts) Convert the two's complement number $(1001\ 0100)_{2c}$ into decimal notation.

Answer: One way to do this is Recognize the number is negative. Negate it to get $-(1001\ 0100)_{2c} = (0110\ 1100)_{2c}$. Use Horner's rule to convert $(0110\ 1100)_{2c}$ to decimal

Horner's Rule @ $x = 2$

0	1	1	0	1	1	0	0
	0	2	6	12	26	54	108
0	1	3	6	13	27	54	$\boxed{108}$

Therefore, $(1001\ 0100)_{2c} = -108$. Alternatively, just use Horner's rule on the original number

Horner's Rule @ $x = 2$

1	0	0	1	0	1	0	0
	2	4	8	18	36	74	148
1	2	4	9	18	37	74	$\boxed{148}$

and subtract $2^8 = 256$ from the result

$$(0110\ 1100)_{2c} = 148 - 256 = -108$$

3 (10 pts) Convert the decimal integer -37 into two's complement notation.

Answer: Use repeated remaindering

Repeated Remaindering mod 2

QUOTIENTS	37	18	9	4	2	1
REMAINDERS	1	0	1	0	0	1

to find that *unsigned* 37 is $(10\ 0101)_2$. Therefore, positive *signed* $+37$ is $(0\ 10\ 0101)_2$. Finally, negate this to represent

$$-37 = (1\ 01\ 1011)_2$$

4 Using the ideas explained in class, answer the following questions about the floating point number $(0\ 010\ 1101)_{fp}$.

4.1 (5 pts) Is the value of the number positive or negative?

Answer: The leading (sign) bit is 0, which implies the value is positive (note the value is not 0).

4.2 (5 pts) What is the (decimal) value of the exponent?

Answer: The exponent is written in bias notation and is equal to $(010)_{\text{bias}=3} = 2 - 3 = -1$.

4.3 (5 pts) What is the decimal value of the fractional part?

Answer: The fractional part is

$$(0.1101)_2 = 13/16$$

Note that binary $(1101)_2$ is 13 and because there are four places to the right of the point, you must divide 13 by 16.

4.4 (5 pts) What is the (decimal) value of the normalized floating point number $(0\ 010\ 1101)_{fp}$?

Answer: Using the above answers

$$(0\ 010\ 1101)_{fp} = +\left(1 + \frac{13}{16}\right) \times 2^{-1} = \frac{29}{32}$$

5 (10 pts) The ASCII character set lies first in the Unicode alphabet: A block known as Basic Latin. The largest (last) Latin-1 value $(7F)_{16}$ represents the "delete" character. What is the decimal value of $(7F)_{16}$?

Answer: Use the fact that $(F)_{16} = 15$ and Horner's rule you can convert $(7F)_{16}$ to decimal

Horner's Rule @ $x = 16$

7	15
	112
7	$\boxed{127}$

Therefore, $(7F)_{16} = 127$.

6 The octal alphabet is $\mathbb{O} = \{0, 1, 2, 3, 4, 5, 6, 7\}$.

6.1 (10 pts) How many octal numerals are needed to write 73?

Answer: It takes

$$\lfloor \log_8 73 \rfloor + 1 = 2 + 1 = 3 \quad \text{octal numerals}$$

Alternatively you can see that

$$73 = 64 + 8 + 1 = (111)_8$$

6.2 (5 pts) How many octal strings with fixed-length n are there?

Answer: There are 8^n strings with length n.

6.3 (5 pts) How many non-empty, variable-length octal strings of length n or less are there?

Answer: There are 8 strings of length 1, 8^2 strings of length 2 up to 8^n strings of with length n. You want to find a formula for

$$8 + 8^2 + 8^3 + \cdots + 8^{n-1} + 8^n$$

Name the sum S and note

$$\begin{aligned} 7S &= 8S - S \\ &= 8(8 + 8^2 + 8^3 + \cdots + 8^{n-1} + 8^n) - (8 + 8^2 + 8^3 + \cdots + 8^{n-1} + 8^n) \\ &= (8^2 + 8^3 + 8^4 + \cdots + 8^n + 8^{n+1}) - (8 + 8^2 + 8^3 + \cdots + 8^{n-1} + 8^n) \\ &= 8^{n+1} - 8 \end{aligned}$$

Therefore, the sum is (the number of non-empty, variable-length octal strings of length n or less) is

$$S = \frac{8^{n+1} - 8}{7}$$

Note that $S(n)$ is a function, and generates the sequence

$$0, 8, 72, 584, \ldots$$

7 Consider the arithmetic sequence

$$\vec{X} = \langle 3,\ 7,\ 11,\ 15,\ \ldots,\ 4n-1,\ 4n+3, \ldots \rangle$$

and the quadratic sequence

$$\vec{Y} = \langle 0,\ 3,\ 10,\ 21,\ 36,\ \ldots,\ 2n^2+n, \ldots \rangle$$

7.1 (5 pts) What is the difference sequence $\triangle\vec{Y}$ of $\vec{Y}$ and how is it related to $\vec{X}$?

Answer: The difference sequence $\triangle\vec{Y}$ is the sequence of differences of successive terms in $\vec{Y}$. That is,

$$\triangle\vec{Y} = \langle 3-0,\ 10-3,\ 21-10,\ 36-15,\ \ldots \rangle = \langle 3,\ 7,\ 11,\ 15,\ \ldots \rangle = \vec{X}$$

7.2 (5 pts) What is the function (formula) that computes the sum

$$\sum_{k=0}^{n-1} (4k+3)$$

Answer: The sum telescopes.

$$\begin{aligned}
\sum_{k=0}^{n-1} (4k+3) &= \sum_{k=0}^{n-1} \left(2(k+1)^2 + (k+1) - (2k^2+k)\right) \\
&= (3-0) + (10-3) + (21-10) + (36-21) + \\
&\qquad \cdots + (2n^2+n) - (2(n-1)^2 + (n-1)) \\
&= 2n^2 + n
\end{aligned}$$

Alternatively,

$$\begin{aligned}
\sum_{k=0}^{n-1} (4k+3) &= 4\sum_{k=0}^{n-1} k + 3n \\
&= 4\frac{n(n-1)}{2} + 3n \\
&= 2n^2 + n
\end{aligned}$$

Back to the quiz on page 325.
Back to the notes on page 170.

Keys to Quiz(zes) on Counting

Counting Truth Assignments and Boolean Functions (Key)

1 How many truth assignments are there on 2, 4, 8, 16 and n Boolean variables?

Answer: There are 2 choices for each of the variables.

- For 2 variables there are $2 \times 2 = 2^2 = 4$ truth assignments.
- For 4 variables there are $2 \times 2 \times 2 \times 2 = 2^4 = 16$ truth assignments.
- For 8 variables there are $2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 2^8 = 256$ truth assignments.
- For 16 variables there are $2^{16} = 65536$ truth assignments.
- For n variables there are 2^n truth assignments.

2 How many Boolean functions are there on 2, 4, 8, 16 and n Boolean variables?

Answer: There are 2^k truth assignments on k variables. Each of these can map to one of 2 values.

- For 2 variables there are $2^{2^2} = 16$ Boolean functions
- For 4 variables there are $2^{2^4} = 2^{16}$ Boolean functions
- For 8 variables there are $2^{2^8} = 2^{256}$ Boolean functions
- For 16 variables there are $2^{2^{16}} = 2^{65536}$ Boolean functions
- For n variables there are 2^{2^n} Boolean functions

Some authors use 0, 1, and X for `False`, `True`, and `Don't Care`. See (Patterson and Hennessy, 1993).

3 In implementing Boolean logic, there are situations where we don't care what the value of an input or output is. That is, one of 3 values (`False`, `True`, and `Don't Care`) can be on an input or output line.

3.1 For input don't cares, but only `False` and `True` output, how many Boolean functions on n variables are there?

Answer: There are 3^n input combinations and 2 outputs for each of these. Therefore there are 2^{3^n} Boolean functions of this type.

3.2 For output don't cares, but only `False` and `True` input, how many Boolean functions on n variables are there?

Answer: There are 2^n input combinations and 3 outputs for each of these. Therefore there are 3^{2^n} Boolean functions of this type.

3.3 For both input and output don't cares, how many Boolean functions on n variables are there?

Answer: There are 3^n input combinations and 3 outputs for each of these. Therefore there are 3^{3^n} Boolean functions of this type.

Back to the quiz on page 326.
Back to the notes on page 175.

Counting Functions (Key)

1 Use theorem to show there are 2^{2^n} Boolean functions on n Boolean variables.

Answer: The domain is $\mathbb{X} = \mathbb{B}^n$ and has cardinality 2^n. The range is $\mathbb{Y} = \mathbb{B}$ and has cardinality 2. By the theorem there are

$$|\mathbb{Y}|^{|\mathbb{X}|} = 2^{2^n} \quad \text{functions from } \mathbb{X} \text{ to } \mathbb{Y}.$$

2 How many functions $f : \mathbb{X} \to \mathbb{Y}$ can be defined when the domain and co-domain have the following sizes?

2.1 $|\mathbb{X}| = 2$, $|\mathbb{Y}| = 4$

Answer: There are

$$|\mathbb{Y}|^{|\mathbb{X}|} = 4^2 = 16 \quad \text{functions}$$

2.2 $|\mathbb{X}| = 4$, $|\mathbb{Y}| = 2$

Answer: There are

$$|\mathbb{Y}|^{|\mathbb{X}|} = 2^4 = 16 \quad \text{functions}$$

2.3 $|\mathbb{X}| = 4$, $|\mathbb{Y}| = 8$

Answer: There are

$$|\mathbb{Y}|^{|\mathbb{X}|} = 8^4 = 2^{12} = 4096 \quad \text{functions}$$

2.4 $|\mathbb{X}| = 8$, $|\mathbb{Y}| = 4$

Answer: There are

$$|\mathbb{Y}|^{|\mathbb{X}|} = 4^8 = 2^{16} = 65536 \quad \text{functions}$$

Back to the quiz on page 326.
Back to the notes on page 179.

Counting Subsets (Key)

1 In how many ways can you choose 5 elements from the bits $\mathbb{B}$?

Answer: Well, is this a trick question? You cannot choose 5 elements if there are only 2?.

2 In how many ways can you choose 5 elements from the octal numerals $\mathbb{O}$?

Answer: The set $\mathbb{O} = \{0, 1, 2, 3, 4, 5, 6, 7\}$ contains 8 elements. Therefore, 5 elements can be chosen from $\mathbb{O}$ in

$$\binom{8}{5} = \frac{8 \cdot 7 \cdot 6}{3 \cdot 2 \cdot 1} = 56 \quad \text{ways}$$

3 In how many ways can you choose 5 elements from the set of digits $\mathbb{D}$?

Answer: There are 10 elements in $\mathbb{D}$, Therefore, 5 elements can be chosen from $\mathbb{D}$ in

$$\binom{10}{5} = \frac{10 \cdot 9 \cdot 8 \cdot 7 \cdot 5}{5 \cdot 4 \cdot 3 \cdot 2 \cdot 1} = 252 \quad \text{ways}$$

4 In how many ways can you choose 5 elements from the set of hexadecimal digits $\mathbb{H}$?

Answer: There are 16 elements in $\mathbb{H}$, Therefore, 5 elements can be chosen from $\mathbb{H}$ in

$$\binom{16}{5} = \frac{16 \cdot 15 \cdot 14 \cdot 13 \cdot 12}{5 \cdot 4 \cdot 3 \cdot 2 \cdot 1} = 4368 \quad \text{ways}$$

$|\mathbb{X}| = n$

5 Let $\mathbb{X}$ be an n-element set. How many subsets does $\mathbb{X}$ have?

Answer: $\mathbb{X}$ has 2^n subsets. To construct a subset: For each element x chose to put x in or leave it out of the subset.

6 Why does the sum of values in row n of Pascal's triangle sum to 2^n?

Answer: The values in a row count the number of 0-element, 1-element, 2-element, ... and n-element subsets. The total number of subsets is 2^n. Therefore the sum of values in row n of Pascal's triangle is 2^n

$$\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{n} = \sum_{k=0}^{n} \binom{n}{0} = 2^n$$

7 Let $\mathbb{X}$ be an n-element set. Suppose there are i subsets of $\mathbb{X}$ with k-elements and j subsets of $\mathbb{X}$ with $k-1$ elements. Let y be an element that is not in $\mathbb{X}$. How many k-element subsets of $\mathbb{X} \cup \{y\}$ are there.

Answer: This is just an awkward way to express Pascal's indentity. The number to be computed is

$$\binom{n+1}{k} \quad \text{number of } k\text{-element subsets of an } n\text{-element set.}$$

Given

$$i = \binom{n}{k} \quad \text{and} \quad j = \binom{n}{k-1} = j$$

Pascals's indentity yields

$$\binom{n+1}{k} = \binom{n}{k} + \binom{n}{k-1} = i + j$$

Back to the quiz on page 326.
Back to the notes on page 192.

Counting Relations (Key)

1 How many relations $x \sim y$ can be defined between elements $x \in \mathbb{X}$ and $y \in \mathbb{Y}$ when the domain and co-domain have the following sizes?

1.1 $|\mathbb{X}| = 2$, $|\mathbb{Y}| = 4$
Answer: There are

$$2^{|\mathbb{X}||\mathbb{Y}|} = 2^{2\cdot 4} = 2^8 = 256 \quad \text{relations}$$

1.2 $|\mathbb{X}| = 4$, $|\mathbb{Y}| = 2$
Answer: There are

$$2^{|\mathbb{X}||\mathbb{Y}|} = 2^{4\cdot 2} = 2^8 = 256 \quad \text{relations}$$

1.3 $|\mathbb{X}| = 4$, $|\mathbb{Y}| = 8$
Answer: There are

$$2^{|\mathbb{X}||\mathbb{Y}|} = 2^{4\cdot 8} = 2^{32} \quad \text{relations}$$

1.4 $|\mathbb{X}| = 8$, $|\mathbb{Y}| = 4$
Answer: There are

$$2^{|\mathbb{X}||\mathbb{Y}|} = 2^{8\cdot 4} = 2^{32} \quad \text{relations}$$

1.5 $|\mathbb{X}| = n$, $|\mathbb{Y}| = m$
Answer: There are

$$2^{|\mathbb{X}||\mathbb{Y}|} = 2^{nm} \quad \text{relations}$$

2 Pretend you have 40 Facebook friends. In how many different ways can they "like" each other?

Answer: There are $40 \times 40 = 1600$ ordered pairs of friends (x, y) where, potentially, "x likes y." So there are 2^{1600} possible ways friends could like each other, from no one likes anyone else to everyone likes everyone.

Back to the quiz on page 327.
Back to the notes on page 183.

Counting Perumations Cycles and Partitions (Key)

1 Use the recurrence relation

Answer:

Back to the quiz on page 327.
Back to the notes on page 196.

Keys to Quiz(zes) on Induction

Basic induction problems (Key)

1 Write a function f that maps each integer n to some integer that has a remainder of 3 when divided by 4.

Answer:

$$f(n) = 4n + 3$$

Recall mod 4 partitions the integers into 4 subsets:

$$\{4k : k \in \mathbb{Z}\}, \quad \{4k+1 : k \in \mathbb{Z}\}, \quad \{4k+2 : k \in \mathbb{Z}\}, \quad \{4k+3 : k \in \mathbb{Z}\}$$

2 Use mathematical induction to prove:

$$\sum_{k=0}^{n-1} (4k+3) = n(2n+1)$$

Answer: First, a *basis*: For $n = 1$, the value of the sum is 3, which matches the formula (function) $f(1) = 1(2 \cdot 1 + 1) = 3$.

For the inductive step, state an *assumption* or *hypothesis*:

$$3 + 7 + 11 + \cdots + (4n-1) = \sum_{k=0}^{n-1} (4k+3) = n(2n+1) \quad \text{for some } n \geq 1.$$

And, then draw a *conclusion*: When you add in the next term you get

$$\begin{aligned}
[3 + 7 + 11 + \cdots + (4n-1)] + (4n+3) &= \left[\sum_{k=0}^{n-1} (4k+3)\right] + (4n+3) \\
&= [n(2n+1)] + (4n+3) \\
&= 2n^2 + 5n + 3 \\
&= (n+1)(2n+3) \\
&= f(n+1)
\end{aligned}$$

3 Use mathematical induction to prove the largest n-numeral octal number is $8^n - 1$.

Answer: First, a *basis*: For $n = 1$, the largest octal number is $7 = 8^1 - 1$. The *hypothesis* is, for some $n \geq 1$, the largest n-numeral octal number is $8^n - 1$. And, then the *conclusion* is this: Construct the largest $n+1$ octal number multiplying $8^n - 1$ by 8 and adding 7. That is

$$8 \cdot (8^n - 1) + 7 = 8^{n+1} - 8 + 7 = 8^{n+1} - 1$$

Shifting all numerals left one position is multiplication by the base

$$8 \cdot (777)_8 = (7770)_8$$

This is a common low-level way to multiply by the base.

Back to the quiz on page 327.
Back to the notes on page 209.

Basic problems on recursion (Key)

1 Consider binary strings without consecutive 1's. For instance,

$$\epsilon,\ 0,\ 1,\ 00,\ 01,\ 10,\ 000,\ 001,\ 010,\ 100,\ 101, \ldots$$

Let be s is a string without consecutive 1's.

- how you can extend s to a string without consecutive 1 by appending a single bit.

Back to the quiz on page 327.
Back to the notes on page 218.

Basic problems on recursion (Key)

1 Show that the linear function $f(n) = 4n + 3$ satifies the recurrence

$$f_n = f_{n-1} + 4 \quad \text{with base case } f_0 = 3$$

Answer: For $n = 0$ the function value matches the base case

$$f(0) = 4 \cdot 0 + 3 = f_0 = 3$$

Assume $f(n-1) = 4(n-1) + 3 = f_{n-1}$ for some $n \geq 1$ and substitute it into the equation

$$\begin{aligned} f_n &= f_{n-1} + 4 \\ &= [4(n-1) + 3] + 4 \\ &= 4n + 3 \end{aligned}$$

2 Show that the quadratic function $s(n) = n^2$ satifies the recurrence

$$s_n = s_{n-1} + (2n - 1) \quad \text{with base case } s_0 = 0$$

Answer: For $n = 0$ the function value matches the base case

$$s(0) = 0^2 = s_0 = 0$$

Assume $s(n-1) = (n-1)^2 = s_{n-1}$ for some $n \geq 1$ and substitute it into the equation

$$\begin{aligned} s_n &= s_{n-1} + (2n - 1) \\ &= (n-1)^2 + (2n - 1) \\ &= n^2 - 2n + 1 + (2n - 1) \\ &= n^2 \end{aligned}$$

3 What well-known list of natural numbers has partial sums equal to $s(n) = n^2$?

Answer: The sum of odd numbers are the squares

$$1 = 1^2,\ 1 + 3 = 2^2,\ 1 + 3 + 5 = 3^2,\ 1 + 3 + 5 + 7 = 4^2, \ldots$$

Back to the quiz on page 328.
Back to the notes on page 230.

Summative exam #4 (Key)

1 (10 pts) How many truth assignments and how many Boolean functions can be defined on n Boolean variables?

Answer: There are 2^n truth assignments and 2^{2^n} Boolean functions.

2 The following questions are about a set characters from the animated television series SpongeBob SquarePants. The set is called $\mathbb{SBSP}$ and the characters are: SpongeBob, Patrick, Squidward, Gary the Snail, and Captain Krabs. Let $\mathbb{SP}$ be the set without SpongeBob.

$\mathbb{SBSP} = \{$  $\}$ $\qquad$ $\mathbb{SP} = \{$  $\}$

2.1 (10 pts) How many functions can be defined from $\mathbb{SBSP}$ to $\mathbb{SP}$?

Answer: There are 4^5 functions.

2.2 (10 pts) Is there a one-to-one function from $\mathbb{SBSP}$ to $\mathbb{SP}$? Explain your answer.

Answer: There is not! Mapping more elements to fewer means that some character in $\mathbb{SP}$ is the image of more than one character in $\mathbb{SBSP}$.

2.3 (10 pts) Is there an onto function from $\mathbb{SBSP}$ to $\mathbb{SP}$? Explain your answer.

Answer: There is! One such function would map every character to itself and map SpongeBob to any character you like.

2.4 (10 pts) In how many ways can you arrange (permute) the characters in $\mathbb{SBSP}$?

Answer: There are $5! = 120$ ways to permute the five different characters.

2.5 (10 pts) How many relations can be defined between $\mathbb{SBSP}$ and $\mathbb{SP}$?

Answer: There are $2^{4\times 5} = 2^{20}$ relations.

2.6 (10 pts) How many subsets does $\mathbb{SBSP}$ have?

Answer: There are 2^5 subsets.

2.7 (10 pts) How many subsets of $\mathbb{SBSP}$ contain 3 characters? Give your answer as a natural number.

Answer: There are

$$\binom{5}{3} = \frac{5!}{3!2!} = \frac{5\cdot 4}{2} = 10 \quad \text{subsets with 3 characters.}$$

3 (10 pts) Use mathematical induction to prove:

$$0 + 1 + 2 + 3 + \cdots + (n-1) = \frac{n(n-1)}{2}$$

Answer: The *basis* for induction can start at $n = 1$. In this case, there is only one term in the sum: 0. And, the function is

$$\frac{1(1-1)}{2} = 1$$

Both sides of the identity are equal.

The *inductive step* has two parts.

Assume

$$0 + 1 + 2 + 3 + \cdots + (n-1) = \frac{n(n-1)}{2}$$

for some value of $n \geq 1$.

Prove the sum of the first $(n+1)$ natural numbers equals $(n+1)n/2$. To do so, note that the sum of the first $(n+1)$ natural numbers is the sum of the first n plus the $(n+1)^{\text{st}}$ natural number. This leads to the derivation:

$$\begin{aligned}
[0 + 1 + 2 + 3 + \cdots + (n-1)] + n &= \frac{n(n-1)}{2} + n \\
&= \frac{n(n-1)}{2} + \frac{2n}{2} \\
&= \frac{(n^2 - n) + 2n}{2} \\
&= \frac{(n^2 + n)}{2} \\
&= \frac{(n+1)n}{2} \\
&= \mathrm{T}(n+1)
\end{aligned}$$

4 (10 pts) Use mathematical induction to prove:

$$1 + 2 + 4 + \ldots + 2^{n-1} = 2^n - 1$$

Answer: The proof has these steps:

The *basis* for induction can start at $n = 1$. In this case, the sum is simply $2^0 = 1$, and the function value is $2^1 - 1 = 1$ also. The sum is equal to the value of the function at $n = 1$.

5 *Assume* the value of the sum (of the first n powers of 2) is equal to the value of the function $2^n - 1$, for some $n \geq 1$. That is,

$$(\exists n \in \mathbb{N},\ n \geq 1)(2^0 + 2^1 + 2^2 + 2^3 + \cdots + 2^{n-1} = 2^n - 1)$$

Prove the sum (of the first $(n+1)$ powers of 2) is the same as the value $2^{n+1} - 1$. To do so, note the sum of the first $(n+1)$ powers of 2 is the sum of the first n powers of 2 plus the $(n+1)^{\text{st}}$ power of 2. This leads to the derivation:

$$\begin{aligned}
(2^0 + 2^1 + 2^2 + 2^3 + \cdots + 2^{n-1}) + 2^n &= (2^n - 1) + 2^n \\
&= 2 \cdot 2^n - 1 \\
&= 2^{n+1} - 1
\end{aligned}$$

Back to the quiz on page 329.
Back to the notes on page 241.

Keys to Quiz(zes) on Orders

Order Basics (Key)

1 Prove that divides on the natural numbers is a partial order.

Answer: Let a, b, c and d be natural numbers.

- Divides is reflexive: for all natural numbers a, a divides a because $a = a \cdot 1$.
- Divides is antisymmetric: for all natural numbers a and $b \neq 0$, if a divides b and b divides a, then there exists natural numbers c and d such that

$$b = ac\,(a \text{ divides } b) \quad \text{and} \quad a = bd\,(b \text{ divides } a)$$

  Therefore

$$b = ac = (bd)c \quad \text{therefore } 1 = cd \text{ and both } c = 1 \text{ and } d = 1, \text{ that is } a = b$$

  Note if $b = 0$ then $a = 0$ also since b divides a.
- Divides is transitive: for all natural numbers a, b and c, if a divides b and b divides c, then there exists natural numbers d and e such that

$$b = ad\,(a \text{ divides } b) \quad \text{and} \quad c = be\,(b \text{ divides } c)$$

  Therefore

$$c = be = (ad)e = a(de) \quad \text{therefore } a \text{ divides } c$$

2 Prove that subset on the power set of set is a partial order.

Answer: Let $\mathbb{X}$, $\mathbb{Y}$, $\mathbb{V}$ and $\mathbb{W}$ be subsets of some universal set $\mathbb{U}$ '.

- Subset is reflexive: for all subsets $\mathbb{X}$, $\mathbb{X}$ is a subset of $\mathbb{X}$ because every element $x \in \mathbb{X}$ is in $\mathbb{X}$.
- Subset is antisymmetric: for all subsets $\mathbb{X}$ and $\mathbb{Y}$, if $\mathbb{X}$ is a subset of $\mathbb{Y}$ and $\mathbb{Y}$ is a subset of $\mathbb{X}$, then $\mathbb{X} = \mathbb{Y}$.
- Subset is transitive: for all subsets $\mathbb{X}$, $\mathbb{Y}$, and $\mathbb{V}$ if $\mathbb{X}$ is a subset of $\mathbb{Y}$ and $\mathbb{Y}$ is a subset of $\mathbb{V}$, then $\mathbb{X}$ is a subset of $setY$.

3 All other qualities being equal, would you chose to use a sorting algorithm whose running time behaved like n^2 or one that behaved like $n \lg n$? Explain your answer.

Answer: Since $\lg n < n$ for all $n = 1$, choose an $n \lg n$ algorithm.

Back to the quiz on page 330.
Back to the notes on page 249.

Counting orders by their properties (Key)

1 How many reflexive relations can be defined on sets with cardinality 2, 8, 10, 16, and n?

Answer: The function that counts reflexive relations is $r(n) = 2^{n(n-1)}$. For $n = 2,\ 8,\ 10, and 16$ these values are

$$2^2, \quad 2^{56}, \quad 2^{90}, \quad 2^{240}$$

2 How many antisymmetric relations can be defined on sets with cardinality 2, 8, 10, 16, and n?

Answer: The function that counts antisymmetric relations is $a(n) = \left(2\sqrt{3^{n-1}}\right)^n$. For $n = 2,\ 8,\ 10, and 16$ these values are

$$(2\cdot\sqrt{3})^2 = 12, \quad (2\cdot\sqrt{3^7})^8 = 2^8 3^{28}, \quad (2\cdot\sqrt{3^9})^{10} = 2^{10}3^{45}, \quad (2\cdot\sqrt{3^{15}})^{16} = 2^{16}3^{120}$$

Back to the quiz on page 330.
Back to the notes on page 253.

Keys to Quiz(zes) on Equivalences

Basics about equivalences (Key)

1 Let $\mathbb{X}$, $\mathbb{Y}$ and $\mathbb{V}$ be subsets of a universal set $\mathbb{U}$. Demonstrate the following equivalences by filling in the give truth tables.

1.1 The identity law for intersection: $\mathbb{X} \cap \mathbb{U} = \mathbb{X}$

Input		Output
$x \in \mathbb{X}$	$x \in \mathbb{U}$	$x \in \mathbb{X} \cap \mathbb{U}$
0	1	0
1	1	1

1.2 De Morgan's laws for the set complement of a union: $\neg(\mathbb{X} \cup \mathbb{Y}) = \neg\mathbb{X} \cap \neg\mathbb{Y}$

Input		Output	
$x \in \mathbb{X}$	$x \in \mathbb{Y}$	$x \in \neg(\mathbb{X} \cup \mathbb{Y})$	$x \in \neg\mathbb{X} \cap \neg\mathbb{Y}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

1.3 The Currying law for sets: $(\neg\mathbb{X} \cup \mathbb{Y}) \cap (\mathbb{X} \cup \mathbb{V}) \subseteq (\mathbb{Y} \vee \mathbb{V})$

Input			Output	
$x \in \mathbb{X}$	$x \in \mathbb{Y}$	$x \in \mathbb{V}$	$x \in \neg(\mathbb{X} \cup \mathbb{Y})$	$\neg\mathbb{X} \cup \neg\mathbb{Y}$
0	0	0	1	
0	0	1	1	
0	1	0	0	
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	0	
1	1	1	0	

2 Consider the set of integers

$$\mathbb{Z} = \{0, \pm 1, \pm 2, \pm 3, \pm 4, \pm 5, \ldots\}.$$

and the set of natural numbers

$$\mathbb{N} = \{0, 1, 2, 3, 4, 5, \ldots\}.$$

Let $a, b \in \mathbb{Z}$ and let $n \in \mathbb{N}$. If n divides $a - b$, write

$$a \equiv b \pmod{n}$$

and say a is congruent to b modula n.

3 Show that conguence modulo n is an equivalence relation.

Answer: Congruence is

- Reflexive: $(\forall a \in \mathbb{Z})(a \equiv a \pmod{n})$ because n divides $a - a = 0$.
- Symmetric: $(\forall a, b \in \mathbb{Z})((a \equiv b \pmod{n}) \Rightarrow (b \equiv a \pmod{n}))$ because if n divides $a - b$, then n divides $b - a$. That is, if $a - b = n \cdot q$ for some quotient $q \in \mathbb{Z}$, then $b - a = n(-q)$.
- Transitive: $(\forall a, b, c \in \mathbb{Z})((a \equiv b \pmod{n}) \wedge (b \equiv c \pmod{n}) \Rightarrow (a \equiv c \pmod{n}))$ because if n divides $a - b$ and n divides $b - c$, then n divides $a - c$. That is, if $a - b = n \cdot q$ for some quotient $q \in \mathbb{Z}$ and $b - c = n \cdot q'$, then $a - c = (a - b) + (b - c) = n(q + q')$, and n divides $a - c$.

Back to the quiz on page 330.
Back to the notes on page 262.

Summative exam #5 (Key)

1 (10 pts) Show that the function $t(n) = n(n-1)/2$ satisfies the recurrence equation and initial condition

$$t_n = t_{n-1} + (n-1), \quad t_0 = 0$$

Answer: First, see that the function matches the initial condition.

$$t(0) = \frac{0(0-1)}{2} = 0 = t_0$$

Next, if

$$t(n-1) = t_{n-1} = \frac{(n-1)(n-2)}{2} \quad \text{for some } n \geq 1,$$

then

$$\begin{aligned} t(n) &\stackrel{?}{=} t(n-1) + (n-1) \\ &= \frac{(n-1)(n-2)}{2} + (n-1) \\ &= \frac{(n-1)(n-2) + 2n - 2}{2} \\ &= \frac{n^2 - n}{2} \\ &= \frac{n(n-1)}{2} \end{aligned}$$

establishing that the equation is satisfied. This recurrence generates the Triangular sequence

$$\vec{T} = \langle 0, 0, 1, 3, 6, 10, 15, 21, 28, 36, 45, \ldots \rangle$$

which is the third column in Pascal's triangle.

2 (10 pts) Prove that $t(n) = n \lg n$ satisfies the recurrence

$$t_{2n} = 2t_n + 2n, \; t_1 = 0, \quad \text{for } n = 1, 2, 4, 8, \ldots$$

Answer: First, see that the function matches the initial condition.

$$t(1) = 1 \lg 1 = 0 = t_1$$

Next, if

$$t(n) = t_n = n \lg n \quad \text{for some } n \geq 1,$$

then

$$\begin{aligned} t(2n) &\stackrel{?}{=} 2t(n) + 2n \\ &= 2n \lg n + 2n \\ &= 2n(\lg n + 1) \\ &= 2n(\lg n + \lg 2) \\ &= 2n \lg 2n \end{aligned}$$

establishing that the equation is satisfied.

3 (35 pts) Consider the divides relation $(a \mid b)$ on the set of natural numbers.

3.1 Does 7 divide 38? Explain your answer.

Answer: No, 38 divided by 7 leaves a remainder of 3. 38 is not a multiple of 7.

3.2 Is divides reflexive?

Answer: Since $a = a \cdot 1$, a divides a for every natural number a. That is, divides is reflexive.

3.3 Is divides antisymmetric? Explain your answer.

Answer: If $a \mid b$ and $b \mid a$, then $ac = b$ for some natural number c and $bd = a$ for some natural number d. Therefore, $acd = bd = a$ showing that $cd = 1$ and $c = d = 1$, and $a = b$.

3.4 Is divides symmetric? Explain your answer.

Answer: No. for a counterexample, 7 divides 14, but 14 does not divide 7.

3.5 Is divides transitive? Explain your answer.

Answer: If $a \mid b$ and $b \mid c$, then $ad = b$ for some natural number d and $be = c$ for some natural number e. Therefore, $a(de) = be = c$ showing that a divides c.

3.6 Is divides an equivalence? Explain your answer.

Answer: No. it is not symmetric.

3.7 Is divides a partial order? Explain your answer.

Answer: Yes. it is reflexive, antisymmetric, and transitive.

4 (35 pts) Consider congruence mod m relation ($a \equiv b \pmod{m}$) on the set of integers.

4.1 What does it mean to say that a is congruent to b mod m?

Answer: It means that m divides $a - b$, or $a - b$ is a multiple of m, or a and b have the same remainder when divided by m, or $a = b + mc$ for some integer c.

4.2 Is $a \equiv a \pmod{m}$ for every integer a? Explain your answer.

Answer: Yes. $a - a = 0$ is a multiple of m.

4.3 If $a \equiv b \pmod{m}$ is $b \equiv a \pmod{m}$ for every pair of integers a and b? Explain your answer.

Answer: Yes. If m divides $a - b$, then m divides $b - a$.

4.4 If $a \equiv b \pmod{m}$ and $b \equiv a \pmod{m}$ does $a = b$ for every pair of integers a and b? Explain your answer.

Answer: No. a counterexample is $3 \equiv 8 \pmod{5}$ and $8 \equiv 3 \pmod{5}$, but $3 \neq 8$.

4.5 If $a \equiv b \pmod{m}$ and $b \equiv c \pmod{m}$ is $a \equiv c \pmod{m}$ for every triple of integers a, b and c? Explain your answer.

Answer: Yes.

4.6 Is congruence mod m an equivalence? Explain your answer.

Answer: No. it is not symmetric.

4.7 Is congruence mod m a partial order? Explain your answer.

Answer: No. it is reflexive, antisymmetric, and transitive.

5 (10 pts) What function $f(n)$ satisfies the recurrence $f_n = 2f_{n-1} + 1$ and initial condition $f_0 = 0$? Listing the first few terms generated by the recurrence should help you identify $f(n)$.

Answer: The function is $f(n) = 2^n - 1$.

Back to the quiz on page 331.
Back to the notes on page 271.

Keys to Quiz(zes) on Modular Numbers

Basic cryptography concepts (Key)

1 Consider the modular recurrence equation

$$r_n = (4r_{n-1} + 2) \bmod 7.$$

with seed $r_0 = 1$.

1.1 What sequence is generated?
Answer: The sequence is

$$\langle 1, 6, 5, 1, 6, \ldots \rangle$$

1.2 What is the *period* of the sequence?

2 Pretend you use a Caesar cipher with shift $k = 6$ to send and receive messages.

2.1 What is the encryption of "mrwatsoncomehere?"

2.2 What is decryption of the message "'igxvkjoks?"

"So it goes" is a frequent refrain in Kurt Vonnegut's novel, Slaughterhouse-Five, or The Children's Crusade: A Duty-Dance with Death

3 Pretend you use an affine cipher

$$(5x + 7) \bmod 26$$

to send and receive messages. Encrypt the message "soitgoes"

Back to the quiz on page 333.
Back to the notes on page 276.

Modular arithmetic (Key)

1 Verify the following:

1.1 Verify that 2 is the remainder when 37 is divided by 7. Conclude that 37 can be reduced to 2 mod 7: $37 = 2 \bmod 2$.
Answer: $37 = 5 \cdot 7 + 2$: The remainder is $r = 2$ and the quotient is 5 when 37 is divided by 7. Multiples of 7 are equivalent to 0 mod 7. Therefore, $37 = 2 \bmod 7$.

1.2 Verify that 5 is the (smallest non-negative) remainder when -37 is divided by 7. Conclude that -37 can be reduced to 5 mod 7.
Answer: $-37 = -6 \cdot 7 + 5$. Multiples of 7 are equivalent to 0 mod 7. Therefore, $-37 = 5 \bmod 7$.

1.3 Do you see how you can use $-37 + 37 = 0$ together with $37 = 2 \bmod 7$ to deduce that $-37 = 5 \bmod 7$?

2 Verify the given value of x satisfies the linear congruence.

2.1 $x = 5$ satisfies $3x = 1 \bmod 7$, that is $3^{-1} = 5 \bmod 7$.

2.2 $x = 5$ satisfies $5x = 1 \bmod 8$, that is $5^{-1} = 5 \bmod 8$.

Back to the quiz on page 333.
Back to the notes on page 280.

The Euclidean algorithm and its extension (Key)

1 Given the factorization of numbers below, identify their greatest common divisor.

1.1 $a = 210 = 2 \cdot 3 \cdot 5 \cdot 7$ and $m = 495 = 3^2 \cdot 5 \cdot 11$

Answer: Include each prime factor the maximum number of times it occurs in *both* numbers. The greatest common divisor is

$$\gcd(210, 495) = 3 \cdot 5 = 15$$

1.2 $a = 147 = 3 \cdot 7^2$ and $m = 392 = 2^3 \cdot 7^2$

Answer: Include each prime factor the maximum number of times it occurs in *both* numbers. The greatest common divisor is

$$\gcd(147, 392) = 7^2 = 49$$

2 Use the Euclidean algorithm to compute the greatest common divisor of the numbers below. Identify the sequence of quotients and remainders that are generated.

2.1 $\gcd(m, a) = \gcd(37, 17)$

Answer:

Dividend	=	Divisor	·	Quotient	+	Remainder
m	=	a	·	q	+	r
37	=	17	·	2	+	3
17	=	3	·	5	+	2
3	=	2	·	1	+	1
2	=	1	·	2	+	0

$\therefore\ \gcd(37, 17) = 1$.

2.2 $\gcd(m, a) = \gcd(165, 42)$

Answer:

Dividend	=	Divisor	·	Quotient	+	Remainder
a	=	m	·	q	+	r
165	=	42	·	3	+	39
42	=	39	·	1	+	3
39	=	3	·	13	+	0

$\therefore\ \gcd(165, 42) = 3$.

Back to the quiz on page 333.
Back to the notes on page 288.

Keys to Quiz(zes) on Proofs

Proofs by contradiction (Key)

1 Mimic Euclid's proof that $\sqrt{2}$ is *irrational* to show that $\sqrt{3}$ is *irrational*.

Answer: Pretend $\sqrt{3}$ is ration so that it can be written as

$$\sqrt{3} = \frac{a}{b}, \quad \text{where} \quad a, b \in \mathbb{Z}, b \neq 0 \; textand \; \gcd(a, b) = 1$$

Then, follow the steps:

$$\begin{aligned} \sqrt{3}b &= a \\ 3b^2 &= a^2 \end{aligned}$$

Therefore, a is a multiple of 3 and a can be written as $a = 3c$ for some integer c.

This gives $3b^2 = a^2 = 9c^2$ or $b^2 = 3c^2$. That is, b is a multiple of 3.

But then both a and b are multiples of 3 which contradicts the hypothesis that $\gcd(a, b) = 1$.

Integers that are not muliples of 3, when squared are not multiples of 3.

If $n = 3k+1$, then $n^2 = 9k^2+6k+1$ is not a multiple of 3.

If $n = 3k+2$, then $n^2 = 9k^2+12k+4$ is not a multiple of 3.

2 Prove there is no largest composite number.

Answer: Pretend there is a largest composite number m. Clearly, $m > 2$. Also, $m+1$ and $m+2$ are larger numbers. One of $m+1$ or $m+2$ is divisible by 2 and therefore composite, contradicting the assumption that m is the largest composite.

Back to the quiz on page 334.
Back to the notes on page 299.

Summative exam #6 (Key)

1 (12 pts) Consider the modular recurrence equation

$$r_n = (4r_{n-1} + 2) \bmod 9.$$

with seed $r_0 = 0$. What are the values of r_1, r_2, r_3?

Answer: The generated sequence has period 9.

$$\langle r_0 = 0, r_1 = 2, r_2 = 1, r_3 = 6, r_4 = 8, r_5 = 7, r_6 = 3, r_7 = 5, r_8 = 4, r_9 = 0, \ldots \rangle$$

Simple equations such as this, with significantly larger parameters than 4, 2 and 9 could be used to generate pseudo-random numbers.

2 (12 pts) Consider -37 divided by 5.

2.1 What is the quotient q and smallest non-negative remainder r?

Answer: $-37 = 5 \cdot (-8) + 3$. The quotient is $q = -8$ and the remainder is $r = 3$.

2.2 Reduce $-37 \bmod 5$ and $37 \bmod 5$ to modular integers in $\mathbb{Z}_5$.

Answer: $-37 \bmod 5 = 3$ and $37 \bmod 5 = 2$. Notice that $(3 + 2) \bmod 5 = 0$.

3 (10 pts) Verify the following.

3.1 Show that $x = 4$ satisfies $3x = 1 \bmod 11$.

Answer: $3 \cdot 4 = 12 = 11 + 1 = 1 \bmod 11$.

3.2 What is $3^{-1} \bmod 11$?

Answer: $3^{-1} = 4 \bmod 11$.

4 (12 pts) Use the Euclidean algorithm to compute the greatest common divisor of 19 and 43.

Answer:

$$\begin{aligned} 43 &= 19 \cdot 2 + 5 \\ 19 &= 5 \cdot 3 + 4 \\ 5 &= 4 \cdot 1 + 1 \\ 4 &= 1 \cdot 4 + 0 \end{aligned}$$

$\therefore \ \gcd(43, 19) = 1$.

5 (12 pts) Find parameters s and t such that

$$43s + 19t = \gcd(43, 19)$$

Answer:

		2	3	1	4	Quotients
1	0	1	3	4	19	
0	1	2	7	9	43	
	+1	−1	+1	−1	+1	Determinants

The last determinant is

$$\det \begin{vmatrix} 4 & 19 \\ 9 & 43 \end{vmatrix} = 43 \cdot 4 - 19 \cdot 9 = 172 - 171 = 1 = \gcd(43, 19)$$

That is, $43s + 19t = 43 \cdot 4 + 19 \cdot (-9) = 1$, where $s = 4$ and $t = -9$.

6 (10 pts) Compute the value of x that solves the linear congruence equation

$$19x = 4 \bmod 43$$

Answer: From the previous problem, you know $19^{-1} = -9 \bmod 43$. Therefore,

$$-9 \cdot 19x = x = -9 \cdot 4 = -36 = 7 \bmod 43$$

As a check, notice that

$$19 \cdot 7 = 133 = 43 \cdot 3 + 4 = 4 \bmod 43$$

7 (12 pts) Use a proof by contradiction to show that $\sqrt[3]{2}$, the cube root of 2, is irrational.

Answer: Assume that $\sqrt[3]{2}$ is rational. Then

$$\sqrt[3]{2} = \frac{a}{b} \quad \text{where } a, b, \in \mathbb{Z} \text{ and } b \neq 0.$$

Moreover, the fraction a/b can be assumed to be in reduced form, that is, $\gcd(a, b) = 1$. Then, follow th equations:

$\sqrt[3]{2} = \frac{a}{b}$		cube both sides
$2 = \frac{a^3}{b^3}$	clear the sides of fractions	
$2b^3 = a^3$		conclude $a = 2k$ is even
$2b^3 = 8k^3$		divide by 2
$b^3 = 4k^3$		conclude b is even

The conclusion that both a and b are even contradicts the assumption. Therefore, $\sqrt[3]{2}$ is irrational.

8 (10 pts) What function $f(n)$ satisfies the recurrence $f_n = f_{n-1} + (n-1)$ and initial condition $f_0 = 0$? Listing the first few terms generated by the recurrence should help you identify $f(n)$.

Answer: The function is $f(n) = n(n-1)/2$.

9 (10 pts) Let $\mathbb{X} = \{a, b, c, d\}$.

9.1 Is $\{\{a, b\}, \{c, d\}\}$ a partition of $\mathbb{X}$? Explain your answer.

Answer: Yes, $\{a, b\}$ and $\{c, d\}$ are two non-empty disjoint subsets whose union is $\mathbb{X}$.

9.2 Is $\{\{a, b, c\}, \{c, d\}\}$ a partition of $\mathbb{X}$? Explain your answer.

Answer: No, $\{a, b, c\}$ and $\{c, d\}$ are not disjoint.

Back to the quiz on page 334.
Back to the notes on page 293.

Mindmaps: Discrete mathematics

Figure 13: Mind map: Discrete mathematics, part 1

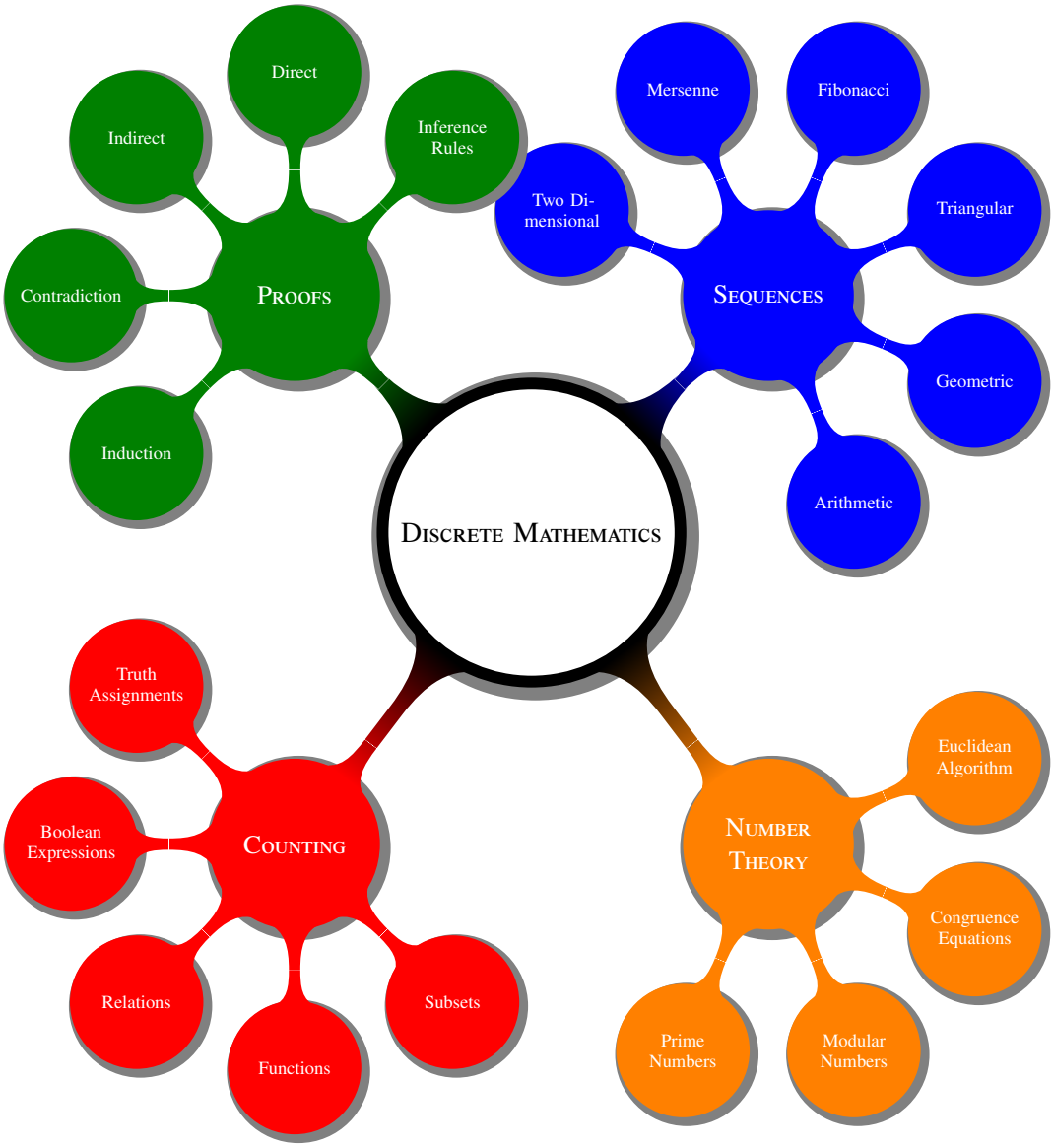


Figure 14: Mind map: Discrete mathematics, part 2

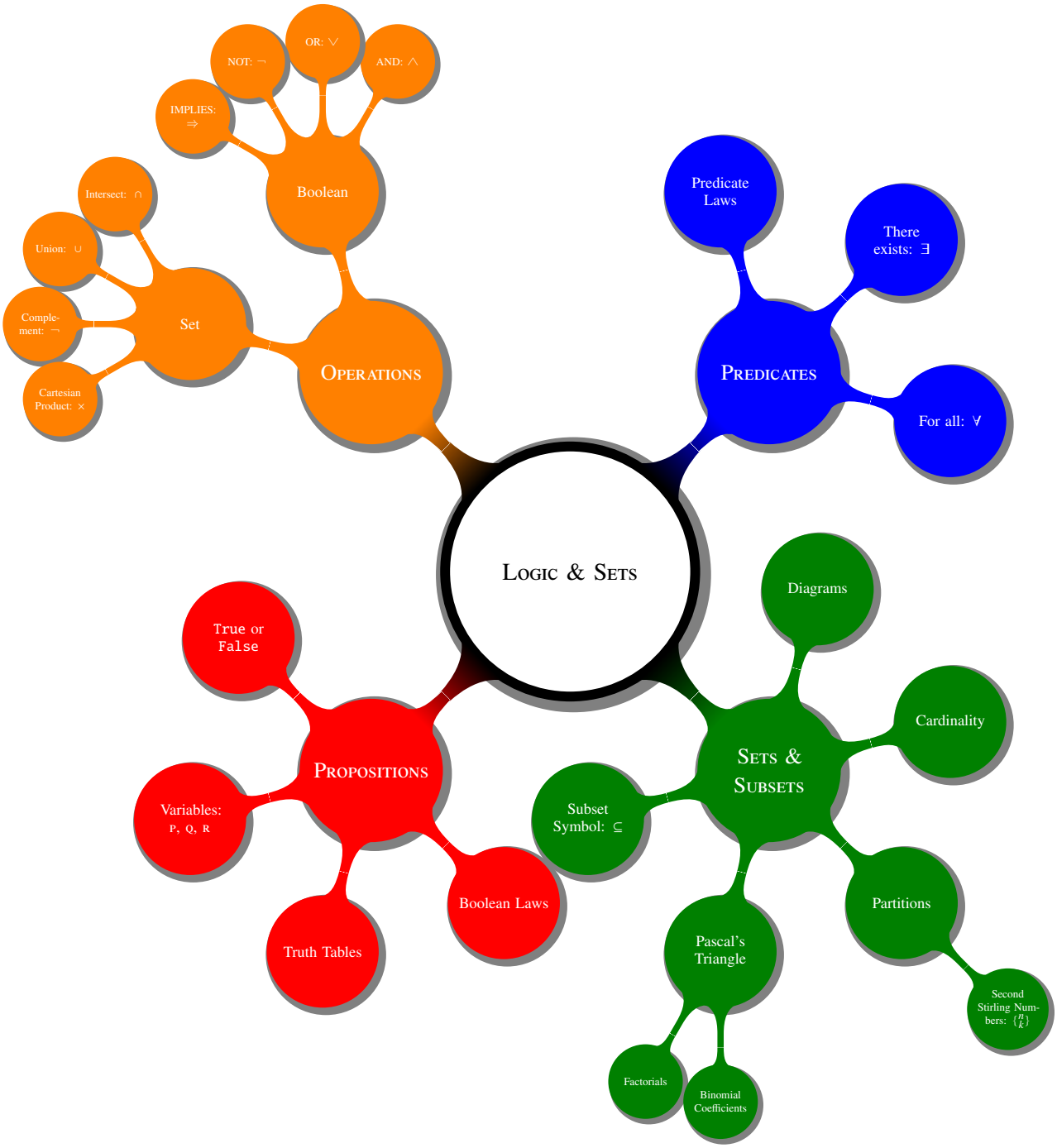


Figure 15: Mind map: Logic and sets

Figure 16: Mind map: Sequences

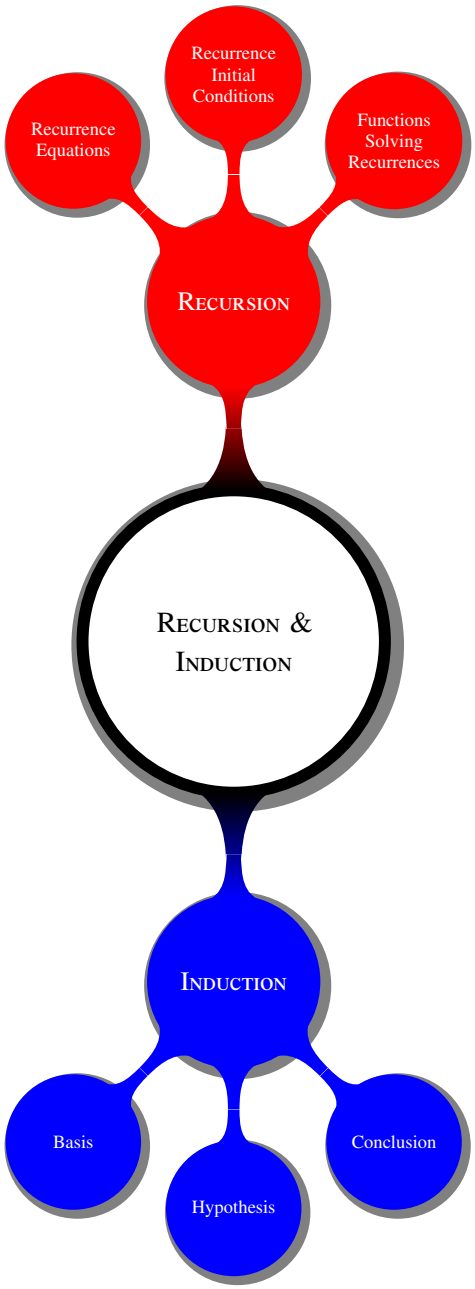


Figure 17: Mind map: Recursion and induction

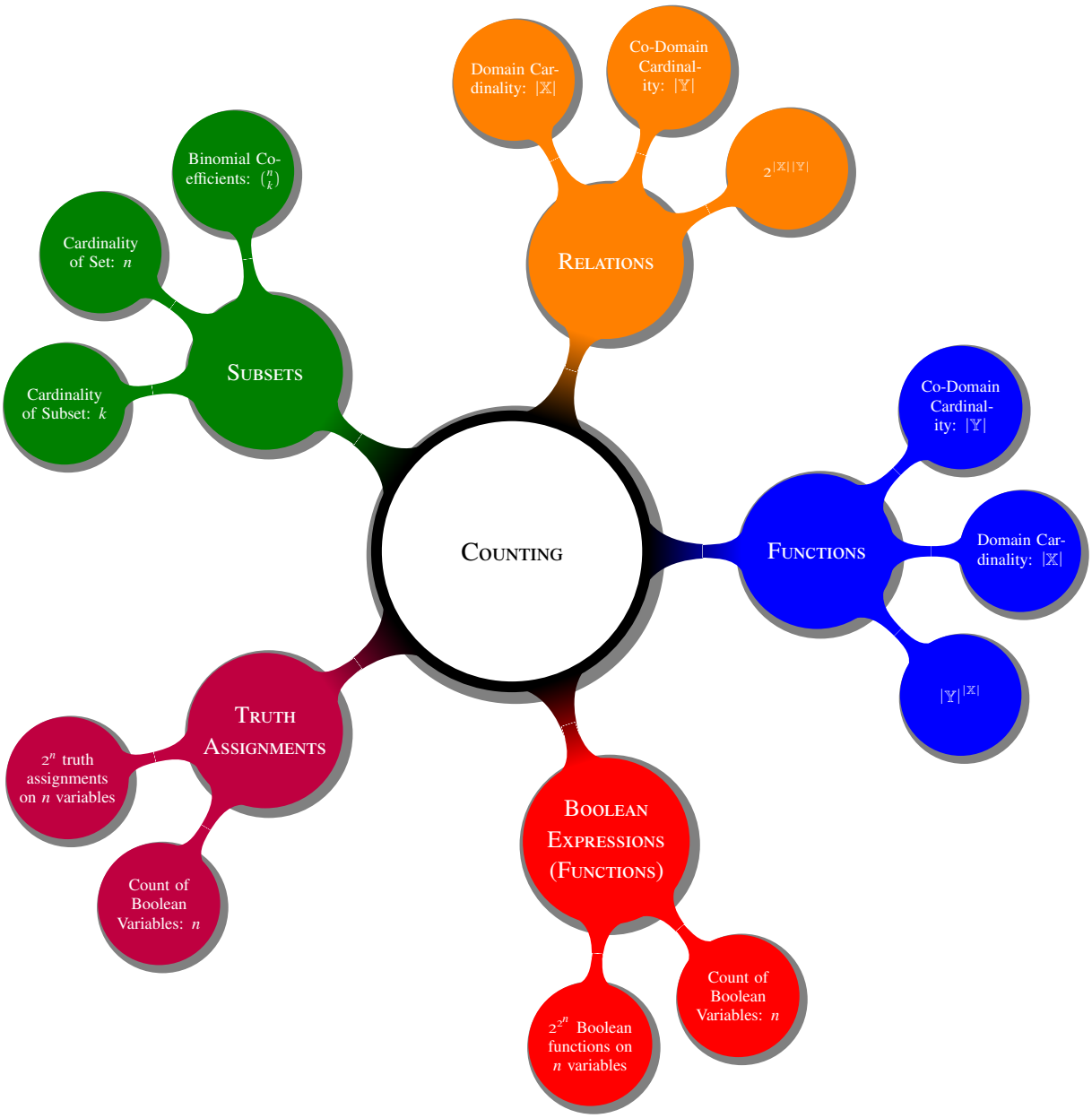


Figure 18: Mind map: Counting

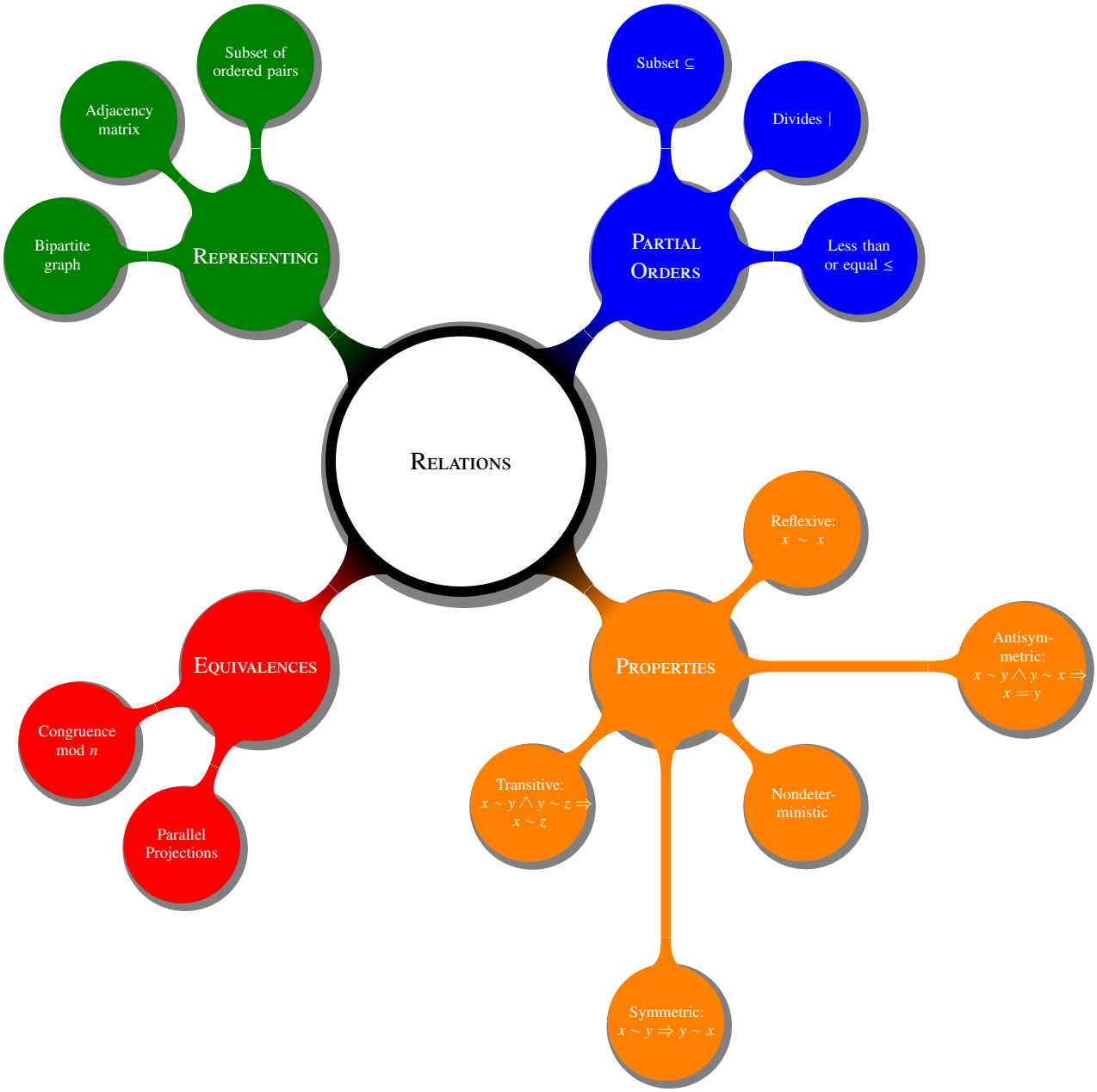


Figure 19: Mind map: Relations

Figure 20: Mind map: Functions

Figure 21: Mind map: Numbers and naming

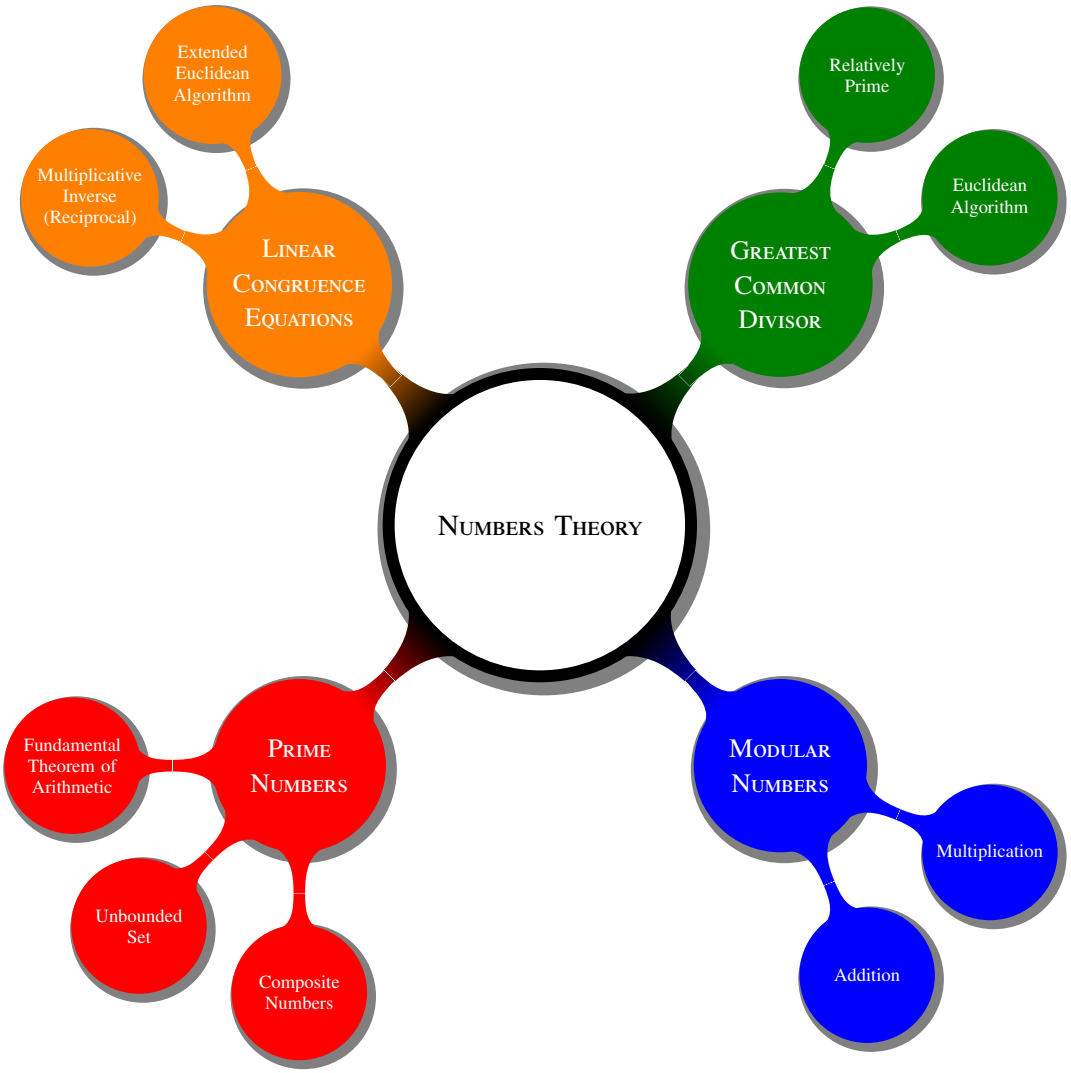


Figure 22: Mind map: Number theory

Bibliography

Belcastro, Sarah-Marie (2012). *Discrete Mathematics with Ducks*. Taylor & Francis.

Bender, E. A. and Williamson, S. G. (2005). *A Short Course in Discrete Mathematics*. Dover.

Calkin, N. and Wilf, H. S. (2000). Recounting the rationals. *The American Mathematical Monthly*, 107:360–363. [page 94]

Carroll, L. (1958). *Symbolic Logic, Part I: Elementary & The Game of Logic*. Dover Publications (originally published by MacMillan and Co., Ltd., London, in 1897 and 1887), New York, N.Y. [page 87]

Carroll, L., Gardner (Editor), M., and Tenniel (Illustrator), J. (2000). *The Annotated Alice: The Definitive Edition*. W. W. Norton. [page 33], [page 47], [page 130], [page 159]

Enzensberger, H. (1999). *Drawbridge up: mathematics–a cultural anathema (Zugbrücke außer Betrieb: Mathematik im Jenseits der Kultur)*. A.K. Peters.

Epp, S. (2010). *Discrete Mathematics with Applications*. Brooks/Cole Publishing Company, 4th edition.

Fuson, K. C., Kalchman, M., and Bransford, J. D. (2005). Mathematics understanding: An introduction. In Donovan, M. S. and Bransford, J. D., editors, *How Students Learn: History, Mathematics, and Science in the Classroom*, pages 217–256. National Academies Press, Washington, D. C. [page 14]

Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA. [page 141]

Graham, R. L., Knuth, D. E., and Patashnik, O. (1989). *Concrete Mathematics*. Addison-Wesley. [page 12], [page 36], [page 109], [page 116]

Gries, D. and Schneider, F. B. (1994). *A Logical Approach to Discrete Mathematics*. Springer-Verlag.

Hayes, B. (2001). How to count. *American Scientist*, 89(2). [page 171], [page 257]

Hayes, B. (2005). Naming names. *American Scientist*, 93(1):6.

Hayes, B. (2006). Gauss's day of reckoning. *American Scientist*, 94(3). [page 131]

Higham, N. J. (1996). *Accuracy and Stability of Numerical Algorithms*. SIAM, Philadelphia, PA, USA. [page 141]

Hofstadter, D. R. (1999). *Gödel, Escher, Bach: An Eternal Golden Braid. 20th Anniversary Edition*. Basic Books, New York, New York. [page 215]

Kaner, C., Falk, J. L., and Nguyen, H. Q. (1999). *Testing Computer Software, Second Edition*. John Wiley & Sons, Inc., New York, NY, USA, 2nd edition. [page 259]

Kelly-Bootle, S. (1981). *The Devil's DP Dictionary*. McGraw-Hill paperbacks. McGraw-Hill. [page 215]

Knuth, D. E. (1997a). *The Art of Computer Programming, Volume 1 (3rd Ed.): Fundamental Algorithms*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA. [page 12]

Knuth, D. E. (1997b). *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA. [page 12]

Knuth, D. E. (1998). *The Art of Computer Programming, Volume 3: (2nd Ed.) Sorting and Searching*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA. [page 12]

Miller, G. A. (1994). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 101(2):342–352. doi: 10.1037/0033-295X.101.2.343. [page 168]

O'Donnell, J., Hall, C., and Page, R. (2006). *Discrete Mathematics Using a Computer*. Springer-Verlag, London, England, second edition.

Papadimitriou, C. H. (1994). *Computational Complexity*. Addison-Wesley. [page 85]

Patterson, D. A. and Hennessy, J. L. (1993). *Computer Organization and Design: The Hardware/Software Interface*. Morgan Kaufmann Publishers, Inc. [page 326], [page 380]

Rosen, K. H. (2011). *Discrete Mathematics and Its Application*. McGraw-Hill, seventh edition. ISBN 9780073383095.

Shannon, C. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656. [page 47]

Smullyan, R. (1985). *To Mock a Mockingbird*. Knopf Doubleday Publishing Group. [page 61]

Stanoyevitch, A. (2011). *Discrete Structures With Contemporary Applications*. A Chapman & Hall book. Chapman and Hall/CRC.

Tufte, E. R. (1990). *Envisioning Information*. Graphics Press, Cheshire, Connecticut. [page 12]

Tufte, E. R. (1997). *Visual Explanations*. Graphics Press, Cheshire, Connecticut. [page 12]

Tufte, E. R. (2001). *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, Connecticut. [page 12]

Tufte, E. R. (2006). *Beautiful Evidence*. Graphics Press, LLC, first edition. [page 12]

Wilkinson, J. H., editor (1988). *The Algebraic Eigenvalue Problem*. Oxford University Press, Inc., New York, NY, USA. [page 141]

Index