Name:

CSE 1400 Applied Discrete Mathematics

Fall 2014 Quiz 3

Score

1. (10 pts) Pretend you want to name $4096 = 8^4$ things using fixed-length octal strings. How long would the strings need to be?

   Answer: You can name m things in octal using strings of length $\lceil \log_8 m \rceil$. In this instance, where $m = 4096$, strings of length $\lceil \log_8 4096 \rceil = 4$ are needed: The names can be $0 = (0000)_8$ to $4095 = (7777)_8 = 7(8^2 + 8^2 + 8 + 1)$.

Score

2. (10 pts) Pretend you needed to write the number 4096 in octal. How many octal numerals are needed?

   Answer: You can write 4096 using $(\lfloor \log_8 4096 \rfloor + 1) = 5$ octal numerals. Note that

$$2^{12} = 8^4 = 4096 \quad \text{so that } \log_8 4096 = 4$$

   and 4096 can be written in

$$\lfloor \log_8 4096 \rfloor + 1 = 5 \quad \text{octal numerals.}$$

   Indeed, $(1\ 0000)_8 = 8^4 = 4096$.

Score

3. (10 pts) Computers were once programmed with Hollerith cards. Hardly anyone numbered their cards even though card decks were frequently dropped and picked up in random order.

   Pretend you were prepared, numbered your cards, dropped them, and picked them up in this order

$$\langle 3, 2, 5, 0, 4, 1 \rangle$$

   What permutation will sort the cards into their original order? Write your answer using cyclic notation. How many cycles are in the permutation?

   Answer: It is useful to start by writing the permutation in matrix form

$$\begin{array}{cccccc|l} 0 & 1 & 2 & 3 & 4 & 5 & \text{normal order} \\ 3 & 2 & 5 & 0 & 4 & 1 & \text{picked-up order} \end{array}$$

   The permutation makes the following map

$$\text{card picked-up} \mapsto \text{where it belongs}$$

$$0 \mapsto 3 \quad 1 \mapsto 2 \quad 2 \mapsto 5 \quad 3 \mapsto 0 \quad 4 \mapsto 4 \quad 5 \mapsto 1$$

   Therefore the permutation is

$$[0, 3][1, 2, 5][4] \quad \text{which has 3 cycles}$$

   The permutation $\langle 3, 2, 5, 0, 4, 1 \rangle$ can be represented in cycle notation as $[0, 3][1, 5, 2][4]$. Apply the permutation $[0, 3][1, 2, 5][4]$ to $[0, 3][1, 5, 2][4]$.

$$\begin{aligned} 0 &\mapsto 3 \mapsto 0 \\ 1 &\mapsto 5 \mapsto 1 \\ 2 &\mapsto 1 \mapsto 2 \\ 3 &\mapsto 0 \mapsto 3 \\ 4 &\mapsto 4 \mapsto 4 \end{aligned}$$

   That is, $[0, 3][1, 2, 5][4]$ sorts $[0, 3][1, 5, 2][4]$.

Score

4. (15 pts) Let $\mathbb{B}^n$ be the set of bit strings of length n, for instance, $\mathbb{B}^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$. Let

$$B(n, k) = \text{"the number strings in } \mathbb{B}^n \text{ with } k \text{ bits set to 0 and the remaining } n-k \text{ bits set to 1"}$$

For instance, $B(3, 2) = 3$ because each of 001, 010, and 100 have exactly 2 bits set to 0 (and no other 3-bit string does).

(a) Look at the sets $\mathbb{B}^1$, $\mathbb{B}^2$, and $\mathbb{B}^3$. Form a table showing the values of

$$\begin{array}{lllll} B(1,0) & B(1,1) & & & 1 \quad 1 \\ B(2,0) & B(2,1) & B(2,2) & & = 1 \quad 2 \quad 1 \\ B(3,0) & B(3,1) & B(3,2) & B(3,3) & 1 \quad 3 \quad 3 \quad 1 \end{array}$$

(b) What is the value of $B(7, 4)$?

Answer: Constructing the table above was meant to let you draw the conclusion that the problem is about binomial coefficients. Notice that $B(n, k)$ can be interpreted as the number of way to choose k places from n and write a 0 there; filling the remaining places with 1. That is $B(n, k) = \binom{n}{k}$.

Use the factorial expression for $B(7, 4)$

$$B(7, 4) = \binom{7}{4} = \frac{7!}{4!3!} = \frac{7 \cdot 6 \cdot 5}{3!} = 35$$

(c) Each string in $\mathbb{B}^n$ ends in either a 0 or a 1. Use this idea to show how $B(7, 4)$ can be computed from $B(6, 4)$ and $B(6, 3)$.

Answer: Understanding that the problem is about binomial coefficients, write Pascal's identity

$$B(7, 4) = B(6, 4) + B(6, 3) \quad \text{or} \quad \binom{7}{4} = \binom{6}{4} + \binom{6}{3}$$

Here, the explanation: Let s be a string with length 7 and 4 bits set to 0. There are two (disjoint) cases:

If s ends in 0, then it was formed from a string of length 6 with 3 bits set to zero.

If s ends in 1, then it was formed from a string of length 6 with 4 bits set to zero.

Therefore, adding these two counts gives

$$B(7, 4) = B(6, 4) + B(6, 3)$$

Score

5. (15 pts) A little genetics. The DNA alphabet is $\mathbb{DNA} = \{A, C, G, T\}$. A codon is a triple xyz where $x, y, z \in \{A, C, G, T\}$, The standard genetic code groups codons into 22 (equivalence) classes called amino acids. For instance, TGG codes for the amino acid Tryptophan.

   (a) How many codons are there?
   Answer: $4^3 = 64$

   (b) Explain why there must be an amino acid with 3 or more codons?
   Answer: If every amino acid had only 2 or less codons, there would be 44 or fewer codons.

   (c) Explain why there must be an amino acid with 2 or less codons?
   Answer: If every amino acid had 3 or more codons, there would be 66 codons, more than 64.

Score

6. (20 pts) Give a brief answer to the following questions.

   (a) How many functions are there from $\mathbb{B}$ to $\mathbb{D}$?
   Answer: There are $|\mathbb{D}|^{|\mathbb{B}|} = 10^2 = 100$ functions.

   (b) How many relations are there from $\mathbb{B}$ to $\mathbb{D}$?
   Answer: There are $2^{|\mathbb{D}| \times |\mathbb{B}|} = 2^{20}$ relations

   (c) What is the sum of the values in row n of Pascal's triangle?
   Answer: The sum is 2^n. The values in row n count the number of subsets of a given size. There sum must equal the total number of subsets.

   (d) 20 people were told to stand in line. In how many ways could they line up?
   Answer: There are 20! permutations (orders) of the people.

Score

7. (10 pts) Use mathematical induction to show that the sum of the first n powers of 2 is equal to $2^n - 1$.

Answer: For no powers, $n = 0$, the sum is empty and equal to 0, the value of $2^0 - 1$. For 1 term, the sum is $2^0 = 1$ which equals $2^1 - 1$.

If the statement is `True` for some n, then consider the sum

$$(1 + 2 + 2^2 + \cdots + 2^{n-1}) + 2^n$$

The first n terms in the above sum is $(2^n - 1)$. Therefore the sum of the first $(n + 1)$ terms is

$$(1 + 2 + 2^2 + \cdots + 2^{n-1}) + 2^n = (2^n - 1) + 2^n = 2^{n+1} - 1$$

Score

8. (10 pts) The time complexity of merge sort is modeled by the initial condition $T(1) = 0$ and recurrence equation

$$T(n) = 2T\left(\frac{n}{2}\right) + n \quad \text{for } n = 2, 4, 8, \ldots$$

Which states: To sort n things partition them into two halves; sorting each half; and merging the results.

Show that the function $T(n) = n \lg n$ satisfies initial condition and the recurrence equation.

Answer: Let $T(n) = n \lg n$. This function satisfies the initial condition

$$T(1) = 1 \lg 1 = 0$$

Next consider the right-hand side of the equation:

$$\begin{aligned} 2T\left(\frac{n}{2}\right) + n &= 2\left[\frac{n}{2} \lg \frac{n}{2}\right] + n \\ &= n \lg \frac{n}{2} + n \\ &= n(\lg n - 1) + n = n \lg n = T(n) \end{aligned}$$

Total Points: 100 Monday, November 3, 2014