Analysis of Algorithms **Comprehensive Examination, Fall 2003**

Sign the exam with your student number - not your name _____

Answer the following questions to the best of your ability.

1. (20 pts) Problem P can be solved by algorithm A_1 with time complexity $T_1(n) = 2^n$ or algorithm A_2 with time complexity $T_2(n) = 4n^2$. Explain which algorithm you would choose to use under various circumstances.

2. (20 pts) Numbers formed from sums often appear in the analysis of algorithms. What is the big-O size of the sums given below:

    1. Powers of 2: $S_n = 1 + 2 + 4 + \cdots + 2^n$

    2. Positive integers: $S_n = 1 + 2 + 3 + \cdots + n$

    3. Fractions: $S_n = 1 + 1/2 + 1/3 + \cdots + 1/n$

    4. Binomial coefficients: $S_n = \binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{n}$

3. (20 pts) The `CYK` algorithm solves the problem:

   "Is string x in the language generated by context-free grammar G?"

   Let G be a context-free grammar with start symbol S and let x be a string over the terminals of G. Pseudocode for CYK is provided below. Analyze the time-complexity of the CYK algorithm. Assume the complexity of *creating sets*, *forming unions* of sets, and *testing membership* of an element in a set can each be done in constant time.

```
boolean CYK(Grammar G, String x) {
  int n = x.length();
  for (int i = 1; i <= n; i++) {
    V_i1 = {A | A → a, where a is the  ith letter in x}
;
  }
  for (int j = 2; j <= n; j++) {
    for (int i = 1; i <= n-j+1; j++) {
       V_ij = ∅;
       for (int k = 1; k <= j-1; j++) {
          V_ij = V_ij ∪ {A | A → BC, B ∈ V_ik, C ∈ V_i+k,j-k}
       }
    }
  }
  if (S ∈ V_1n) return true; else return false;
}
```

4. (25 pts) Assume there is a method in class `BinaryTree` that prints the *value* stored at each node in tree.

```
void treePrint(Node node) {
  if (null != node) {
     treePrint(node.left());
     node.print();
     treePrint(node.right());
  }
}
```

Pretend the tree has n nodes.

1. Best case analysis:
    1. Write a recurrence equation that describes the best case time complexity for `treePrint()`.
    2. What is the solution to this equation?
    3. What is the space complexity in the best case?

2. Worst case analysis:
    1. Write a recurrence equation that describes the worst case time complexity for `treePrint()`.
    2. What is the solution to this equation?
    3. What is the space complexity in the worst case?

3. Average case analysis:
    1. Write a recurrence equation that describes the average case time complexity for `treePrint()`.

5. (15 pts) Write a recursive divide-and-conquer algorithm for computing the product of an array of integers. Analyze its time and space complexity.