Computer Science 477/677 Spring 2001 Examination, March 26, 2001

Name:_____

No books, notes, scratch paper, or calculators. Use pen or pencil, any color. Use the rest of this page and the backs of the pages for scratch paper. If you need more scratch paper, it will be provided.

The entire test is 100 points.

1. Fill in the blanks. [5 points each blank.]

   (a) The median of a list of n numbers can be found in _____ time. (I want an asymptotic answer.)

   (b) At each step, a _____ algorithm makes that choice which gives the maximum immediate benefit.

   (c) In an _____ hash table, collisions are resolved by storing items in unused positions in the table, rather than storing more than one item in a position.

   (d) If a binary tree has n nodes and height h, then we can say, using asymptotic notation, that $h = O(\_\_\_\_\_)$, and also that $h = \Omega(\_\_\_\_\_)$.

2. What is the difference between a stack and a queue? (Only the difference, please. I do not want a definition of either concept.) [5 points]

3. Suppose a hash table is used to store customer records for a business. You access a record by entering the customer name. There are roughly 10000 customers, and the size of the hash table will be 16384. (That is, 2^14.) A good hash function should have certain characteristics. List them.

   I have a very definite list in mind, and all of my items can be described with very few words. I will grade this problem by checking to see if you write the characteristics on my list. I will **subtract** points if you write irrelevant or incorrect characteristics, or if you are too wordy. Do **not** illustrate the characteristics by giving examples, as this is also a test to see if you can express your ideas clearly without using examples. [20 points]

4. Use Huffman's algorithm to find an optimal binary prefix code for the alphabet *A, B, C, D, E, F*, given that these symbols appear with frequencies proportional to the following table: [20 points]

A	*B*	*C*	*D*	*E*	*F*
2	2	3	5	6	9

5. There are finitely many paths in an acyclic directed graph. Describe a dynamic programming algorithm where the input is a directed acyclic graph G, and the output is the number of paths in G of **odd** length. You may assume that the names of the nodes are 1, 2, ..., n, and there is no backwards edge, that is; if there is an edge from i to j, then $i < j$. [30 points]

   Here is what you must write:

   - The set of subproblems.
   - An ordering on the subproblems. (That is, the order in which they are computed.)
   - A description of the computation for a given subproblem, given solutions to the previous subproblems.
   - Computation of the final answer, given solutions to all subproblems.

A path can start or end anywhere. This directed acyclic graph has 13 paths of odd length.