Computer Science 477/677 Spring 1999 Examination, April 7, 1999

Name:____

No books, notes, or scratch paper. 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. True or false. [5 points each.]

   (a) ________ Greedy algorithms are fast, but never give the correct solution.

   (b) ________ If there is an exponential time algorithm for a problem, there is always a way to design a polynomial time algorithm for the same problem, by containing the exponential explosion.

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

   (a) "______________________________ addressing" is the use of unused places in a hash table to resolve collisions.

   (b) Suppose that each position in a hash table contains a search structure holding all items hashing to that position. The hash table has size m and there are n items in the table. If you execute a *find* for an item x that is in the hash table, there may be items other than x in the table that hashed to the same place. What is the average number of those "other" items? ______________________________

3. What is "quadratic probing"? [10 points]

4. What is the difference between a stack and a queue? (There is only one possible answer, and it is very short. I will *take off* points if you write more than necessary.) [10 points]

5. What is a *greedy* algorithm? State an example of a problem where there is a greedy algorithm which is very efficient for solving that problem. [10 points]

6. Design an algorithm for the following problem.

   You are given an unlimited supply of tiles. Each tile has one letter on it and has a point value. If the letter is A, the point value is 1. If the letter is B, the point value is 2. If the letter is C, the point value is 3. If the letter is D, the point value is 1. Each word that can be constructed from these tiles has a values obtained by adding the values of the tiles. For example, the word AABCAD has value 9.

   There are exactly 5 words whose point value is 2, namely AA, AD, B, DA, DD. Design an $O(n)$-time dynamic programming algorithm that computes, for a given integer n, how many words with point value n can be constructed from these tiles. [20 points]

7. Define the ADT *array*. Remember, this is an *abstract* data type. Give an example of a problem where the implementation of an abstract array should *not* be the usual contiguous block of memory. [20 points]

8. There is a very much faster algorithm for the scrabble problem than the obvious $O(n)$-time dynamic programming algorithm that you gave for problem **??**. This faster algorithm uses memoization. Describe it. [20 points]