B1 Algorithms (25 points)

Select any 5 of the following 6 items (do only 5 -- if you do all 6, only the first 5 will be graded). For each item that you select, give a short explanation of the item (data structure, algorithm, or technique). You should capture the main idea/ideas of the data structure, algorithm, or technique in your explanation, and, for algorithms, state the running time.

1) Kruskal's algorithm

2) Strassen's algorithm

3) Red/black trees

4) Dijkstra's algorithm

5) The Boyer-Moore algorithm

6) Fibonacci Heaps

B2 Algorithms (25 points)

Do all three parts.

a. [8 points] In FORTRAN there is an EQUIVALENCE statement. The effect of EQUIVALENCE A, B is that variables A and B share a storage location. When a number of EQUIVALENCE statements appear in a program, the effect is that the variables are broken into equivalence classes based on those EQUIVALENCE statements. For instance, if EQUIVALENCE A, B; EQUIVALENCE A, C; and EQUIVALENCE D, B; appear in a program, then variables A, B, C, and D all share a single location. Describe a very fast method for processing a sequence of EQUIVALENCE statements (ie determining which variables share a location)? Also, state the running time of your method.

b. [9 points] You are given a set of items $\{I_1, I_2, ..., I_n\}$, with each item of size, $0 < I_i < 1$, and an unlimited supply of bins, each of capacity 1. The objective is to pack the items into a minimum number of bins. Here, packing items into bins means assigning each item to a bin such that the sum of the sizes of the items in a bin does not exceed 1.

Since bin packing is NP-hard, an approximation algorithm must be utilized. One such is the Best Fit algorithm, which is as follows:

Definition: An ***open bin*** is a bin that contains at least one item.

In order from the largest item to the smallest item, pack each item I_i as follows:

If I_i will fit into an open bin
then among the open bins where I_i will fit, pack I_i into a bin with the least amount of available space
else pack I_i into a bin that contains no items

Explain what major data structures/algorithms you would use in order to provide a fast implementation of Best Fit. Be sure to explain why the use of those data structures/algorithms is necessary and how you would use that data structure/algorithm.

c. [8 points] Carefully explain how to compute the median of an array of n numbers in linear time.

B3 Algorithms (25 points)

Do all three parts.

a. [5 points] Suppose a and b are two positive integers of length n when written in base B. What is the cost, measured in the number of operations on digits of base B, of the classical (school) method of multiplying two positive integers?

b. [15 points] Suppose that a and b are positive two digit integers in base B. Thus, $a = a_0 + a_1B$ and $b = b_0 + b_1B$ where $0 \le a_0, a_1, b_0, b_1 < B$. Then, the product ab can be written as:

$$ab = a_0b_0 + [(a_0 + a_1)(b_0 + b_1) - a_0b_0 - a_1b_1]B + a_1b_1B^2$$

Use this observation to write a faster algorithm for the multiplication of two numbers, each consisting of n base B digits. How fast is your algorithm? Clearly show your algorithm and clearly explain and justify the recurrence or sum on which you base your asymptotic cost (where such cost is expressed in terms of n, with B being a constant).

c. [5 points] How fast is the asymptotically fastest known algorithm for integer multiplication? Briefly, what is the technique on which it is based?

B4 Algorithms (25 points)

Do all three parts.

a. [6 points] Define P, NP and NP-complete.

b. [6 points] Name and briefly describe four different NP-complete problems (do not use any type of coloring problem).

c. [13 points] The following problem, ***4 coloring*** is NP-complete. Prove it. Note that you may assume that graph 3 coloring is already known to be NP-complete. Note also that the proof technique "restriction" is not acceptable - you need to do a real reduction.

4 coloring: *Instance: An undirected graph G.*
Question: Is G 4-colorable?