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) Amortized analysis

2) Union/Find in time O(n log* n) or better

3) Depth first search

4) Dynamic programming

5) Fast Fourier Transform

6) Strassen's matrix multiplication algorithm

B2 Algorithms (25 points)

Answer all seven parts

a. [6 points] Select any 2 of the following 3 items (do only 2 -- if you do all 3, only the first 2 will be graded). For each tree that you select you should describe: the structure of the tree; the order associated with the items stored in the tree; and how an Insert operation is performed on that tree (if the Insert involves a rotation, you need only say that there is a rotation AND what the rotation is designed to accomplish. You do not have to give the details of the rotation). Do not describe how to perform any operation other than Insert.

- AVL tree
- 2-3 tree
- Red-black tree

b. [1 point] Define binomial trees.

c. [3 points] For binomial heaps, describe: the structure; the order associated with the items stored there; how an Insert operation is performed. Do not describe how to perform any operation other than Insert.

d. [3 points] Describe how to implement a Decrease_Key operation on a binomial heap. Your method should not change the fundamental structure of a binomial heap. What is the running time of your implementation of Decrease_Key?

e. [5 points] Give a proof by induction that the trees that exist in a Fibonacci Heap are binomial trees if the only operations executed in the Fibonacci Heap are Insert, Min and Delete_min.

f. [4 points] In a Fibonacci Heap, state the worst case and amortized running times for these operations: Insert, Min, Delete_min, Union, Decrease_Key.

g. [3 points] In a Fibonacci Heap, in the implementation of Decrease_Key there is a cascading cut. Explain why cascading cuts are performed.

B3 Algorithms (25 points)

Do all three parts.

a. [12 points] For each of the following algorithms, describe the main idea/ideas, and state the running time.

- Counting sort
- Radix sort
- Quicksort

b. [8 points] It is known that the lower bound for any comparison-based sorting algorithm is $\Omega(n \log n)$. This bound, however, does not necessarily hold if more information about the input is available for use. For example, the input may be "semi-sorted" already. Consider the following two cases and determine for each whether the problem can be solved faster than $\Omega(n \log n)$. Justify your answers.

i. The input array is a min-heap.

ii. Each element in the input array is within 5 positions of its sorted position. To be more precise, the rank of element x_i in the input is between $i - 5$ and $i + 5$. (Assume the input integers are distinct.)

c. [5 points] Given a set of n integers, design a linear time algorithm to find and sort the k largest keys, where k is $O(n / \log(n))$.

B4 Algorithms (25 points)

Do both parts.

a. [12 points] Define P, NP, polynomially reducible and NP-complete.

b. [13 points] Give a complete proof that the following problem is NP-complete. You may assume that 3SAT, CLIQUE, PARTITION and SAT are known NP-complete problems.

4SAT (4 Satisfiability)

INSTANCE: Boolean variables x_1, x_2, ..., x_n and a set of clauses C_1, ..., C_k, where each clause is the disjunction of exactly four literals (recall that a literal is a variable or its negation).

QUESTION: Does there exist a truth assignment to the variables such that the conjunction of the clauses is satisfied (i.e. is true)?