COMP 482, Fall 2008, Exam 1

You have **5 contiguous hours** of your choice to look at and work on the exam. Hand it in during class or at my office, DH 3118. Remember that evening and weekend access to my office is limited.

You may use any material provided as part of this course, including your textbook, notes, and graded assignments, but no other materials. You may not use a computer, except to access the course's online material or to type your answers. Please write legibly, clearly label each problem, and show your work.

Since this is a take-home exam, it is not practical to ask for clarifications during the exam. If you feel that a problem is not sufficiently specified, clearly state the assumptions that you are making, and continue with the exam. Of course, your assumptions should be reasonable ones.

The 6 problems follow on separate pages.

1. (10 points)

   Let $f(n)$ and $g(n)$ be asymptotically positive functions. Prove or disprove that $f(n) = O(g(n))$ implies $g(n) = \Omega(f(n))$.

2. (15 points)

   You are given a sorted array $A[1..n]$, in increasing order, and an item x which might or might not be in A. Show that a lower bound for any *comparison-based* algorithm to find the smallest item in A that is larger than x takes $\Omega(\log n)$ time.

3. (30 points – 10 points each part)

   Consider the following randomized strategy for searching for a value x in an unsorted array $A[1..n]$:

   > Pick a random index i into A. If $A[i] = x$, then we terminate; otherwise, we continue the search by picking a new random index into A. We continue picking random indices into A until either we find an index j such that $A[j] = x$ or we have checked every element of A.

   Note that we pick from the whole set of indices each time, so that we may examine a given element more than once.

   (a) Suppose that there is exactly one index i such that $A[i] = x$. What is the expected number of indices into A that must be picked before x is found and the search terminates?

   (b) Generalizing your solution to part (b), suppose that there are $k \geq 1$ indices i such that $A[i] = x$. What is the expected number of indices into A that must be picked before x is found and the search terminates? Your answer should be a function of n and k.

   (c) Suppose that there are no indices i such that $A[i] = x$. What is the expected number of indices into A that must be picked before all elements of A have been checked and the search terminates?

4. (15 points)

   Use induction to prove that radix sort works on a dataset $\{a_{1,1} \cdots a_{1,d}, \ldots, a_{n,1} \cdots a_{n,d}\}$. Where does your proof need the assumption that the intermediate sort is stable?

5. (20 points – 5 points each part)

   (a) What can you say about the shape of the splay tree after splaying on the minimum element?

   (b) How can you simply and efficiently merge two splay trees t_1 and t_2, assuming you know that all the elements in t_1 are smaller than those in t_2?

   (c) How can you simply* and efficiently partition a splay tree into two splay trees t_1 and t_2, such that all elements of t_1 are less than a given value x, and all elements of t_2 are greater than x? Assume that x is in the original splay tree.

   (*) Note: By simply, we mean more simply than the general BST partitioning algorithm which requires a series of merging steps.

(d) *Briefly* explain why the merging and partitioning algorithms take amortized $O(\log n)$ time, where n is the number of data elements.

6. (10 points)

   Show, for any positive integer n, a sequence of Fibonacci-heap operations that creates a Fibonacci heap consisting of just one tree that is a linear chain of n nodes.