Problem Set 3

Assigned: June 10
Due: June 17

Problem 1.

Suppose that you are given the problem of returning in sorted order the k smallest elements in an array of size n, where k is much smaller than $\log(n)$, but much larger than 1.

Describe how quicksort can be adapted to this problem.

Problem 2.

Suppose that we modify the standard definition of a binary search tree to add a field N.size at each node, which records the size of the subtree under N (including N itself).

A. Explain how to modify the procedure for adding an element X to a tree. *Be sure to consider both the case where X is not yet in the tree and is added, and the case where X is already in the tree, and the tree remains unchanged.*

   You only need to describe the *changes* that are made to the standard algorithm; you do not have to repeat the standard algorithm.

B. Explain how to modify the procedure for deleting an element. As in (A), consider both cases.

C. Describe a procedure for finding the kth largest element in the tree.

D. Describe a procedure for finding the number of elements in the tree less than X.

All of these procedures should run in time proportional to the height of the tree.

Problem 3

Suppose you have two binary search trees P and Q. Let $|P|$ and $|Q|$ be the number of elements in P and Q, and let h_P and h_Q be the heights of P and Q. Assume that that is, $h_P \ll h_Q \ll |P| \ll |Q|$ and

A. Give a destructive algorithm for creating a binary search tree containing the union $P \cup Q$ that runs in time $O(|P|^2)$ in the worst case.

B. Assume now that it is known that the largest element of P is less than the smallest element of Q. Give a destructive algorithm for creating a binary search tree containing the union $P \cup Q$ that runs in time h_P.

C. (1 point extra credit). Find a solution to part (A) that runs in time $O(|P| \cdot h_Q)$ in the worst case, and additionally guarantees that the height of the output tree is no greater than $h_P + h_Q$.