Problem Set 6

Assigned: Oct 31
Due: Nov. 7

The usual rules for submitting homework problems with pictures apply to problem 3.B.

Problem 1:

Suppose that we modify the definition of the binary search tree so that, at each node $N$ in addition to the value, it shows the size of each subtree under $N$. The picture below shows an example.

A. Show how you can use this extra data field to implement the method `nth(N)` which returns the $N$th smallest element in the tree. Assume 0-based counting. For instance, in the above tree, T.nth(0) should return 1; T.nth(3) should return 6.

B. Describe how the “Add” method must be modified to maintain this size field accurately. The procedure should continue to work with a running time proportional to the depth of the node being added.

C. Describe how the “Delete” method must be modified to maintain this size field accurately. The procedure should continue to work with a running time proportional to the depth of the deepest of the nodes that are involved.

Problem 2

A. Write the expression “$(4+(2*10))/((7-5)*(1+2))$” in prefix notation.

B. Write the same expression in postfix notation.

C. Give a trace of the execution of the algorithm `evaluatePostfix` operating on your answer to B by stating the sequence of pushes and pops.
Problem 3

Suppose that you have the following heap.

```
A. Show the array implementation of this heap.
B. Suppose that you execute, in sequence, DeleteMin(); DeleteMin(); Add(7). Show the final state of the heap, in tree form.
```

Problem 4

Show a trace of mergesort running on the following array (length = 21)

```
[21, 22, 85, 84, 99, 83, 31, 2, 36, 46, 24, 64, 35, 7, 15, 88, 44, 9, 12, 77, 5]
```

Use the recursive version of mergesort and produce a trace in the style example 1 in the lecture notes. Do not give an internal trace of merge. Assume that the recursion goes down to array length 3.

Problem 5

You are given an unsorted array $A$ of ints with no repeated elements, and asked to find the $K$th largest elements in descending sorted order. For example if $A$ is the array [11,6,1,2,15,7,4,8,20] and $K = 3$, then the answer should be [20,15,11]. Describe how you would modify selection sort and heapsort to solve this problem (two separate answers). What is the worst case running time of your algorithms, as a function of $N = A$.length and $K$?

You may assume that you are describing your algorithm for the benefit of someone who already knows the standard versions of selection sort and heapsort. So you can either write pseudo-code or describe in English how to modify the standard algorithm, whichever you prefer.