

CSE 2320 Lab Assignment 3

Due November 18, 2014

Goals:

1. Understanding of red-black trees.
2. Understanding of recursive binary tree processing.
3. Understanding of subtree *root ranks* in binary search trees for supporting ranking queries.

Requirements:

1. Modify the provided C code for maintaining a red-black tree to process a sequence of commands (standard input) from the following list:

0 - Exit the program

1 x - Insert positive integer key x , unless x is already present in the tree. Besides inserting the key, subtree root ranks must be updated. (Error message if x is a duplicate.)

2 x - Find the rank of x , i.e. the number of keys in the tree that are smaller than x (error message if x is not in the tree).

3 k - Find the key with rank k (error message if k is not between 0 and $n - 1$, inclusive).

4 - Perform an audit on the subtree rank at each node to give a final indication that the tree is “clean” or “corrupt”.

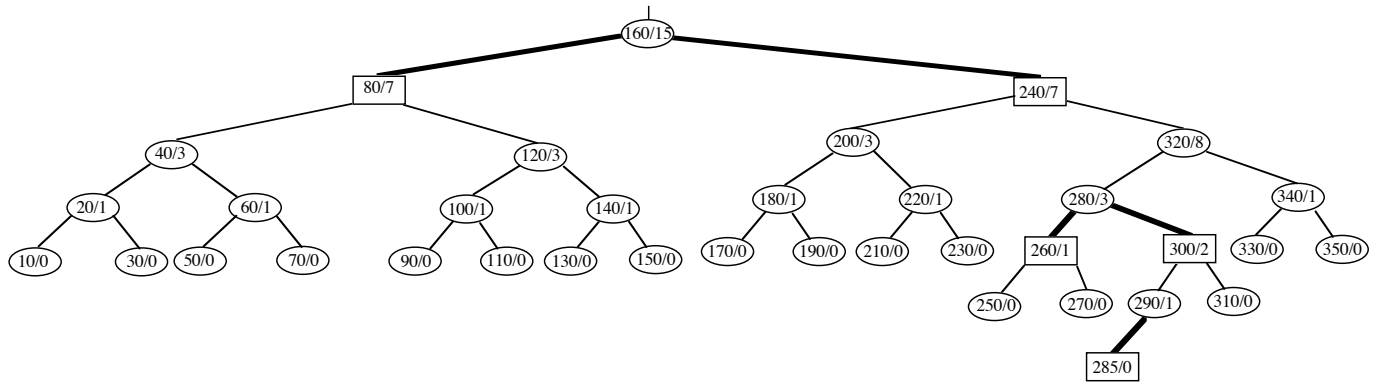
Each command must be echoed to standard output. Commands 1, 2, and 3 must be processed in $\Theta(\log n)$ time.

2. Submit your program source files as a zipped file on Blackboard by 10:45 am on November 18. One of the comment lines should include the compilation command used on OMEGA.

Getting Started:

1. The code changes for maintaining subtree root ranks during command 1 are minor and will include the rotation code.
2. The code for commands 2 and 3 may be coded as variations on recursive search of a binary search tree, but recursion is not required.
3. Command 4 should traverse the tree, compute the rank of the root of each subtree, and verify the stored ranks. Optionally, you could also check 1) the inorder key property, 2) that there is no downward path with consecutive red nodes, and 3) the black-heights are consistent.
4. You are not required to maintain any satellite data related to the integer keys.
5. You may use the driver in the lab directory.

6. An example of a red-black tree with subtree root ranks:



7. The same tree as 6., but using subtree sizes:

