CSCI-311 Challenge Problem 4 (50 points) By Elena AVL-tree

Maybe posted only after Project 2 has been submitted by students (contains "insert" function for AVL-tree)

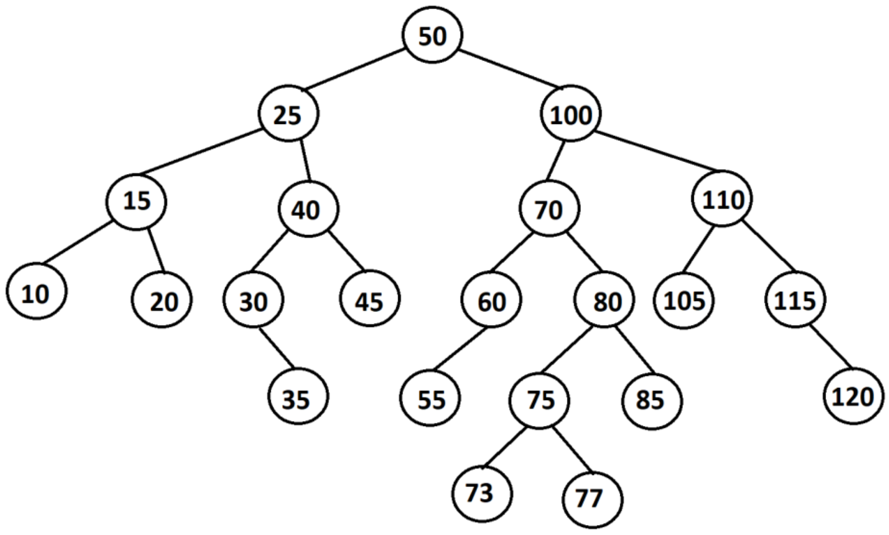
Problem 1 (50 points). ***Find Rank.*** Given a binary search tree of integers, BST, and an integer ***akey***, find the rank of this node in the BST tree and return the rank. Nodes in the BST do not have links to parents.

Note that an integer ***key*** is stored at each node and an integer ***size*** is stored at each node (the total number of nodes in the sub-tree rooted at that node). The ***rank*** is the order of the node when nodes are listed using in-order traversal; the count starts with 0.

Requirements:

1. Your program (functions) must run in O(log(n))-time (if a BST is balanced, then height of the tree is O(log(n))), where *n* is the size of the given BST.
2. Your public function must be called ***findRank*** and take one parameter, the given integer, and return an integer, the rank of the node whose key is given:
   ```
   int findRank(int akey)
   ```
3. If the given value does not exist in the BST tree, then return -1.

Example:



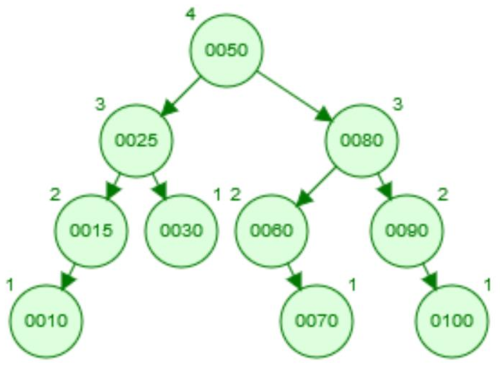
On Figure above, given a node 40, your program will return 6 (there are exactly 6 nodes with values smaller than 40: 10, 15, 20, 25, 30 and 35 of ranks 0, 1, 2, 3, 4, 5. Hence, the rank of 40 is 6).

If the given node is 75, then there are 13 nodes smaller than 75, hence, the rank of 75 is 13.

Submission:

Submit *bst.h* and *bst.cpp* to *challengeProblem4_AVL_FindRank* on **turnin**.

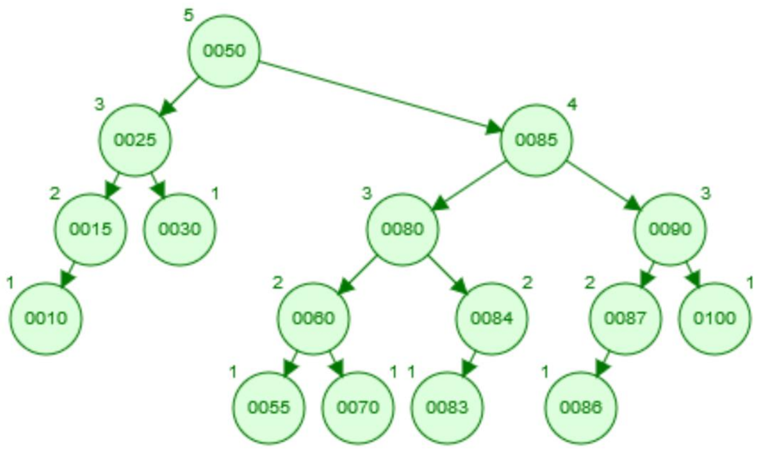
Figure for test files t00.in and t01.in



Given the tree on the left and the key 10, your program will return 0 (if nodes were listed in an array in increasing order, 10 would be at index 0).

Given a key 80, your program will return 7.

Figure for test files t02.in – t07.in:



5
0050
3
0025
4
0085
2
0015
1
0030
3
0080
3
0090
1
0010
2
0060
2
0084
2
0087
1
0100
1
0055
1
0070
1
0083
1
0086