**CS 3050 Quiz # 2, March 22, 2018**

**Time Limit: 75 Minutes**

**Name : KEY Student ID:**

Note: (1) you can use a letter-sized sheet paper with notes; (2) closed-book quiz, no discussion, no use of cell phone; (3) use additional pages or reverse side of quiz pages if needed; (4) all 5 problems have equal number of points.

1. Multiple Choices (circle the correct selection)

(1) Which of the following statements about binary trees is NOT true?

A. Every binary tree has at least one node.

B. Every non-empty tree has exactly one root node.

C. Every node has at most two children.

D. Every non-root node has exactly one parent.

(2) Suppose T is a binary tree with 14 nodes. What is the minimum possible depth of T?

A. 2

B. 3

C. 4

D. 5

(3) What is the special property of red-black trees and what root should always be?

A. a color which is either red or black and root should always be black color only

B. height of the tree

C. pointer to next node

D. a color which is either green or black

(4) What are the operations that could be performed in O(logn) time complexity by red-black tree?

A. insertion, deletion, finding predecessor, successor

B. only insertion

C. only finding predecessor, successor

D. for sorting

5. Which of the following is an application of Red-black trees and why?

A. used to store strings efficiently

B. used to store integers efficiently

C. can be used in process schedulers, maps, sets

D. for efficient sorting

2. Short Answers

(1) Here is a small binary tree:

14

/ \

2 11

/ \ / \

1 3 10 30

/ /

7 40

Write the order of the nodes visited in an in-order traversal:

Answer: 1, 2, 3, 14, 7, 10, 11, 40, 30

(2) While inserting the elements 71, 65, 84, 69, 67, 83 in an empty binary search tree (BST) in the sequence shown, draw the tree it generates:

(3) Is the following statement is true or false? Why?

“The subtree of the root of a red-black tre is always itself a red-black tree.”

Answer:

True as long as the subtree doesn’t violate any properties of the RB tree such as the leaf nodes are all black and any path to the leaf nodes contains the same number of black nodes.

3. Suppose you are given two Binary Search Trees T1 and T2. Present the psuedo code for an efficient algorithm to determine if they are identical trees. [Hint: Identical trees mean that they contain the same elements and have the same tree structure].

Answer:

isSameTree(TreeNode p, TreeNode q)

{

If(p==null && q==null)

Then return true;

Else if(p==null || q==null)

Then return false;

Else

return (p.val == q.val &&

isSameTree(p.left,q.left) &&

isSameTree(p.right,q.right));

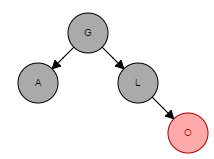
}

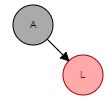
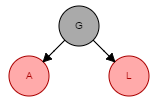
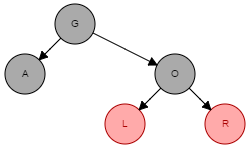
Time Complexity: O(N), Where N is number of nodes in a tree.

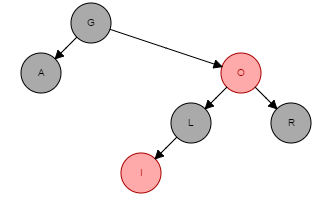
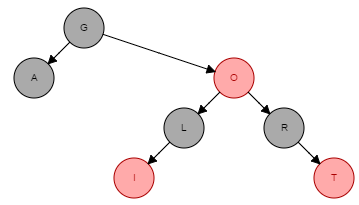
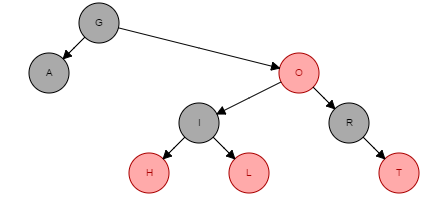
4. Perform Insertion, for each element in Array A, in the Red and Black tree by drawing the tree for each step of insertion.

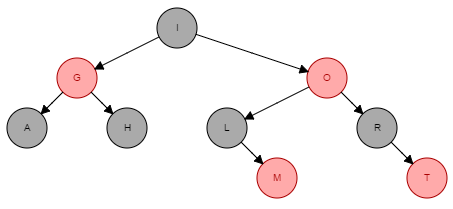
A = {A, L, G, O, R, I, T, H, M}.

Note: The letters are ordered alphabetically. You can use double circles to indicate black nodes, and single circle to indicates red nodes)

Answer:

=>=>=> =>=>

=>=>=>



5. Given an element x in an n-node order-statistic tree and a natural number i, how can we determine the ith successor of x efficiently? [Hint: write a psuedo code to use a combination of OS-SELECT and OS-RANK].

Answer:

Successor(T, x, i)

{

rank=OS-Rank(T,x)

s=rank+i

Node t=OS-Select(T.root, s)

return t

}

Since OS-RANK() and OS-SELECT() both have O(lgn) runtimes, SUCCESSOR() also takes O(lgn).