LEBANESE AMERICAN UNIVERSITY
School of Arts and Science
Department of Computer Science and Mathematics

CSC 310: Algorithms and Data Structures

Lab III

05. Feb. 2016

Implement the class RBNode which represents a binary tree node having an integer value, references to the parent, left child, and right child, and the color. Using RBNode, implement the class RBTree representing a Red-Black tree that will be composed of multiple RBNodes.

In the RBTree class, implement the insert method, which takes as input an integer value and adds it to the tree maintaining the Red-Black tree structure and properties. Use the pseudocode provided to implement the insert and rotation methods.

The Red-Black Tree properties are as follows:

1. Every node is either red or black.
2. The root is black.
3. Every null node is black.
4. If a node is red, then both its children are black.
5. For each node, all simple paths from the node to descendant leaves contain the same number of black nodes.

Pseudocodes

```
LEFT-ROTATE(RBNode x)
1.  y = x.right
2.  x.right = y.left       // turn y's left subtree into x's right subtree
3.  if y.left != y.left
4.        y.left.parent = x
5.  y.parent = x.parent
6.  if x.parent == null
7.        root = y
8.  else if x == x.parent.left
9.        x.parent.left = y
10. else x.parent.right = y
11. y.left = x
12. x.parent = y
```

INSERT(int value)

```
1.  z = new RBNode(value)
2.  y = null
3.  x = root
4.  while x != null
5.      y = x
6.      if z.value < x.value
7.          x = x.left
8.      else x = x.right
9.  z.parent = y
10. if y == null
11.     root = z
12. else if z.value < y.value
13.     y.left = z
14. else y.right = z
15. while z.parent.color == RED
16.     if z.parent == z.parent.parent.left
17.         y = z.parent.parent.right
18.         if y.color == RED
19.             z.parent.color = BLACK
20.             y.color = BLACK
21.             z.parent.parent.color = RED
22.             z = z.parent.parent
23.         else
24.             if z == z.parent.right
25.                 z = z.parent
26.                 LEFT-ROTATE(z.parent.parent)
27.             z.parent.color = BLACK
28.             z.parent.parent.color = RED
29.             RIGHT-ROTATE(z.parent.parent)
30.     else
31.         // same as above but replace left with right and vice versa
32. root.color = BLACK
```

Problem 1

Given a sequence of integers, insert them into a Red-Black tree then print the tree using pre-order traversal.

Input

Your program will be tested against multiple test cases. Each test case is made up of two lines. The first line contains an integer *N* representing the number of integers. The second line has *N* integers to be inserted into the tree.

Output

For each test case, print the tree using pre-order traversal.

Sample Input	Sample Output
7 25 13 10 30 15 27 37	13 10 25 15 30 27 37
4 6 7 8 9	7 6 8 9
6 10 7 15 13 4 6	10 6 4 7 15 13

Problem 2

Given a sequence of integers, insert them into a Red-Black tree then print the height of the tree.

Input

Your program will be tested against multiple test cases. Each test case is made up of two lines. The first line contains an integer *N* representing the number of integers. The second line has *N* integers to be inserted into the tree.

Output

For each test case, print the height of the tree.

Sample Input	Sample Output
7 25 13 10 30 15 27 37	3
4 6 7 8 9	2
6 10 7 15 13 4 6	2