Danny Tan
Dt1462
HW#9

1)
a)
```
insert ( x, Node) {
if  t is null
        set Node to a new Node with element c
if x is less than the element of Node
do recursion of left subtree of Node
else if t is greater than the element of Node
do recursion of right subtree of Node
}
```

b)
```
rangedPrint (low , high, Node) {
if Node is null
return

if the element of Node is greater than low
do recursion of left subtree of Node

if element is in between low and high
print out its element

if the element of Node is less than high
do recursion of right subtree of Node

}
```

c)
```
stringy () {
set num to number of Nodes ( the size of tree)
do an inorder traversal on the root and store all the Nodes into a queueOfNodes
loop through the queue using num
insert the node in the front of queue onto root
pop that Node
}
```
d)

```
averageNodeDepth() {
        return depthSum / numOfNodes
}
```

```
depthSum(Node, depth) {
if Node is null
return 0
endif
return depth  + depthSum for left subtree+  depthSum for right subtree
}

numOfNodes (Node) {
if  Node is null
return 0
endif
return 1 + numOfNodes for left subtree + numOfNodes for right subtree
}
```

2) Average case and worst case is O(n).

3)

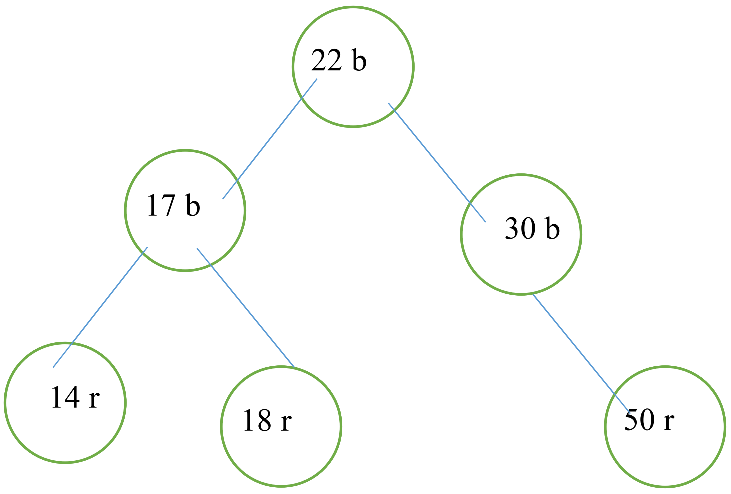
There is nothing wrong with the code. However the t != nullptr is not necessary.

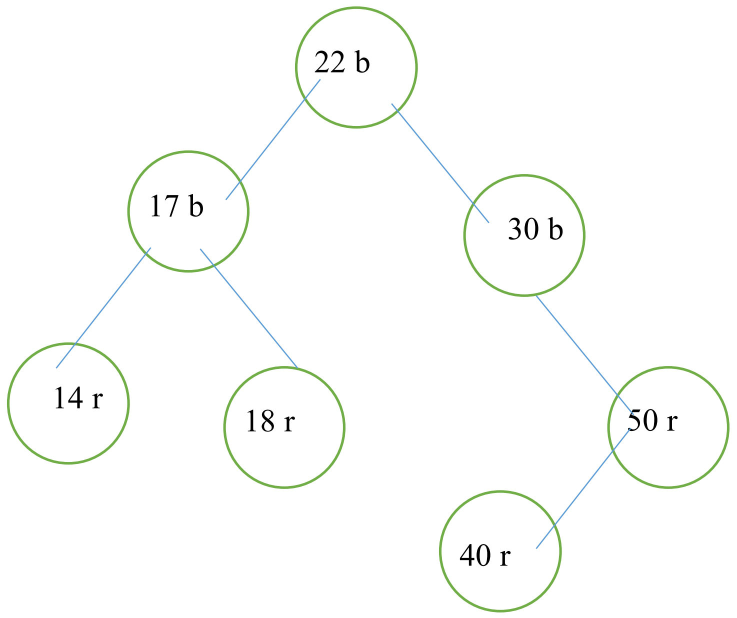
4)

new Node{ x, nullptr, nullptr, 1 } part will make the program not work correctly because at the end of the program, the size is increment again. So the total size of the inserted node, a leaf node will be 2 but that is not correct because its size is suppose to be 1

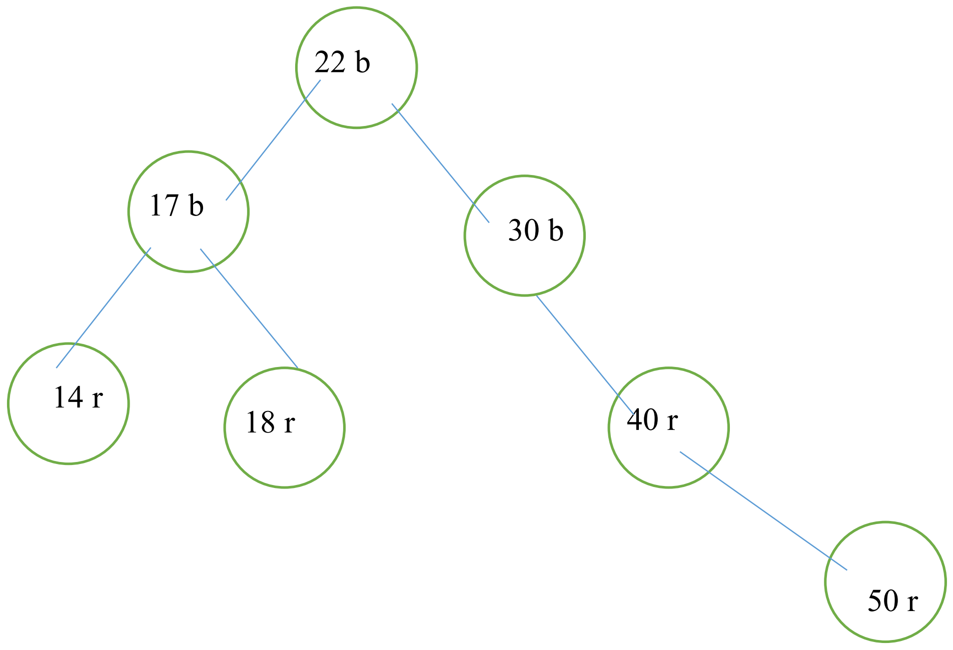
The correct code will be new Node{ x, nullptr, nullptr, 0}

5)

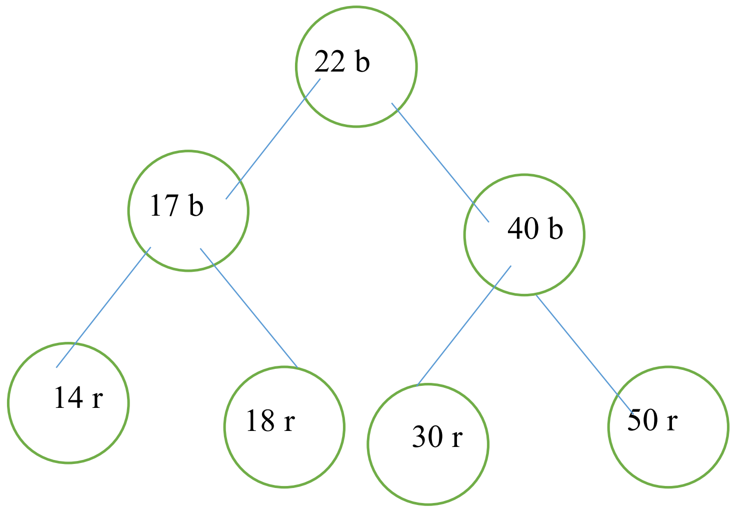




Double red problem case 3. Right Rotate



Double red problem case 2. Left rotate



7)

```
template <class Comparable>
void RedBlackTree::rightRotateRecolor( Node * & k2 ) {
        Node *k1 = k2->left;
        K2->left = k1->right;
        K1->left = k2;
        K2 = k1;
```

8) In the worst case scenario, double red problem case 1 2 and 3 is used. Case 1 does not change the pointer. Case 2 and case 3 both changes 3 pointer. So the maximum pointer change is 6.