

Mohammed touhid chowdhury

Hw9 written

1.

Contains method: first check if its null pointer, if it is you didn't find and you return false. Then check if the node is less than the comparable element if it is then call recursive on the right side. If the x is less than the node, then call the left side. Return true in end so if it's found it will return.

Ranged Print method: if the node is nullptr then return because you reached end. If the elements is greater than low and less than high call recursive on the left and right and print the element. IF the element is greater than high call recursive on the left side. If its less than low than call recursive on the right tree.

Stringy: (couldn't do the code but guessing) call recursive to the left until it hits null. Call recursive on the right until you hit null and then you add to the right of the tree.

Average node depth: in the drive function create a size variable and pass it into the main function. In the main function return if its nullptr (end of tree) and increase the size and then return the depth plus the recursive call for right side and recursive call for left side. Then divide it by size in the driver function

2.

$O(\log n)$

$O(k+h)$

$O(n)$

$O(n)$

3. Yes its valid

4. No not valid, need to pass by reference or your not alias to real one.

5.

```
7. void rightRotateColor(Node*& k2){  
Node* tmp = k2;  
K2 = k2-> left;  
Tmp-left = k2-right;  
Tmp -> color = red;  
K2-> right = tmp;  
K1->color = black;  
}
```

8.

Case one requires no pointer change just recolor. Case 2 we change the grandparents, parents and childs, Case 3 we change parent grand parent and grandparents parent. So total is 6

