MCS 253P Quiz 5

Name: S. ASWIN **ID:** 53844689

1) (70 pts) Consider all the leaves of a binary tree, from left to right order, the values of those leaves form a **leaf value sequence**. For example, in the given trees below, the leaf value sequence is **(6, 7, 4, 9, 8)**. Two binary trees are considered **leaf-similar** if their leaf value sequence is the same. Return **true** if and only if the two given trees with head nodes **root1** and **root2** are leaf-similar.

2) (30 pts) Write one **test case** and **time** and **space complexity**.

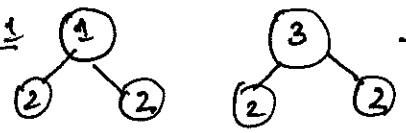
Logic: ① Initialize a ~~map~~ unordered_map.

② For the first tree, find leaf nodes and increment their count in the map.

③ For the second tree, find leaf nodes and decrement their count in the map.

④ Traverse through the map & return true if map value for all key elements are zero, else return false.

T.C 1



→ Both similar Return true

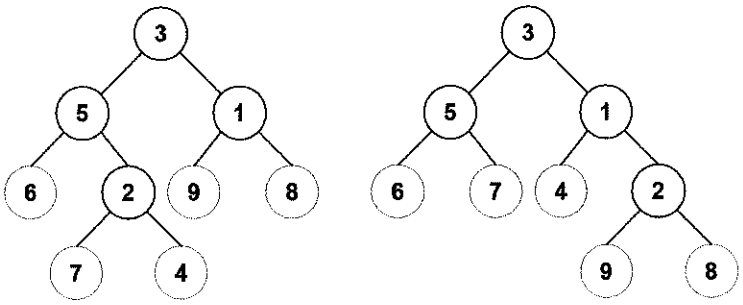
T.C 2

→ Different return false

Time complexity: O(n)

Space complexity: O(n)

```
void findLeafNodes(Node* root, unordered_map<int,int>& um, int value){
    if(!root) return;
    if(root->left==NULL && root->right==NULL)
    {
        um[root->data] += value;
    }
    findLeafNodes(root->left);
    findLeafNodes(root->right);
}
```



Input: root1 = [3,5,1,6,2,9,8,null,null,7,4], root2 = [3,5,1,6,7,4,2,null,null,null,null,null,null,9,8]

Output: true

```
bool isLeafSimilar(Node *root1, Node* root2)
{
    unordered_map<int,int> um;
    findLeafNodes(root1, um, 1);
    findLeafNodes(root2, um, -1);
    for(auto ele : um)
    {
        if(ele.second != 0)
        {
            return false;
        }
    }
    return true;
}
```

```
findLeafNodes(Node *root, unordered_map<int,int>& um, int value)
```