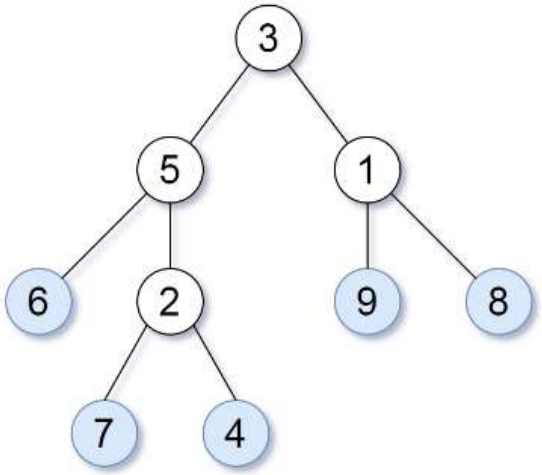
Consider all the leaves of a binary tree, from left to right order, the values of those leaves form a **leaf value sequence**.

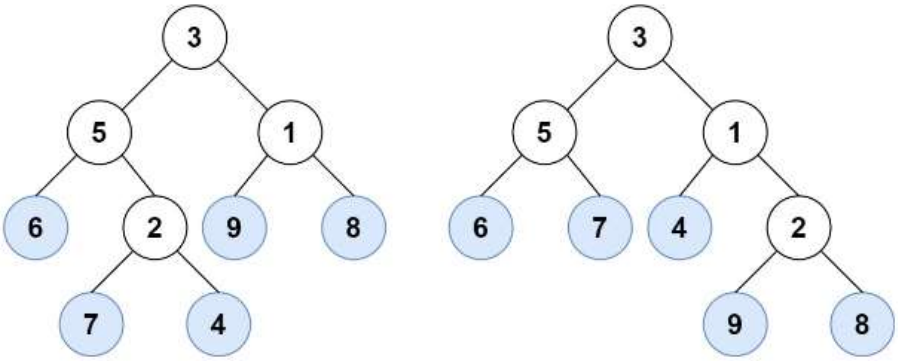


For example, in the given tree above, the leaf value sequence is `(6, 7, 4, 9, 8)`.

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.

Example 1:




```
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
```

Solution:

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode() : val(0), left(nullptr), right(nullptr) {}
 *     TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
 *     TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left),
right(right) {}
 * };
 */
class Solution {
public:
    void getLeaves(TreeNode* tree, vector<int>& vec){
        if (tree){
            getLeaves(tree->left, vec);
            if (!tree->left && ! tree->right)
                vec.push_back(tree->val);
            getLeaves(tree->right, vec);
        }
    }
    bool leafSimilar(TreeNode* root1, TreeNode* root2) {
        vector<int> vec1, vec2;
        getLeaves(root1, vec1);
        getLeaves(root2, vec2);

        return vec1 == vec2;
    }
};
```