

Problem 872: Leaf-Similar Trees

Problem Information

Difficulty: Easy

Acceptance Rate: 70.16%

Paid Only: No

Tags: Tree, Depth-First Search, Binary Tree

Problem Description

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,9,8]` **Output:** `true`

Example 2:

Input: `root1 = [1,2,3]`, `root2 = [1,3,2]` **Output:** `false`

****Constraints:****

* The number of nodes in each tree will be in the range `[1, 200]`. * Both of the given trees will have values in the range `[0, 200]`.

Code Snippets

C++:

```
/**
 * 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:
    bool leafSimilar(TreeNode* root1, TreeNode* root2) {

    }
};
```

Java:

```
/**
 * Definition for a binary tree node.
 * public class TreeNode {
 *     int val;
 *     TreeNode left;
 *     TreeNode right;
 *     TreeNode() {}
 *     TreeNode(int val) { this.val = val; }
 *     TreeNode(int val, TreeNode left, TreeNode right) {
 *         this.val = val;
 *         this.left = left;
 *         this.right = right;
 *     }
 * }
```

```

* this.right = right;
* }
* }
*/
class Solution {
public boolean leafSimilar(TreeNode root1, TreeNode root2) {

}
}

```

Python3:

```

# Definition for a binary tree node.
# class TreeNode:
# def __init__(self, val=0, left=None, right=None):
# self.val = val
# self.left = left
# self.right = right
class Solution:
def leafSimilar(self, root1: Optional[TreeNode], root2: Optional[TreeNode])
-> bool:

```