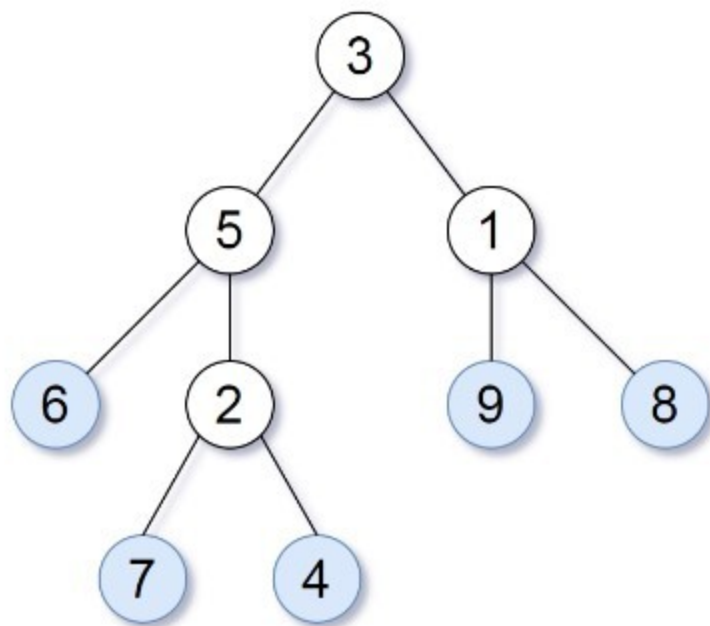


872. Leaf-Similar Trees

Easy Topics Companies

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

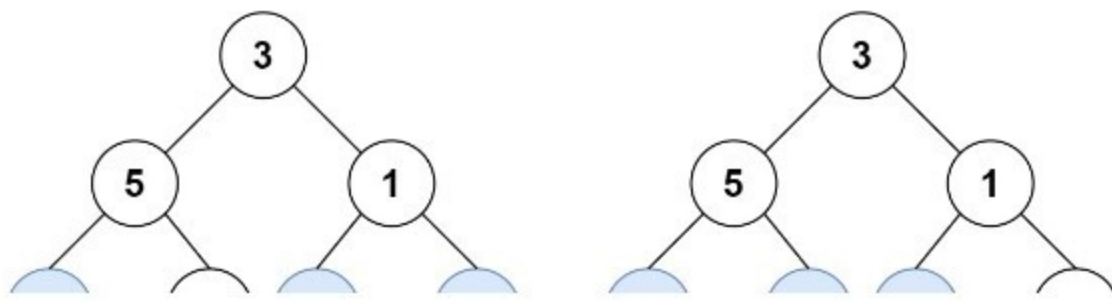


For example, in the given tree above, the leaf value sequence is (6, 7, 4, 9, 8).

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:



Code

C Auto

```

1  /**
2   * Definition for a binary tree node.
3   * struct TreeNode {
4   *     int val;
5   *     struct TreeNode *left;
6   *     struct TreeNode *right;
7   * };
8   */
9  void getLeafSequence(struct TreeNode* root, int *index, int* leaves) {
10     if (root == NULL) {
11         return;
12     }
13
14     if (root->left == NULL && root->right == NULL) {
15         leaves[*index++] = root->val;
16     }
17
18     getLeafSequence(root->left, index, leaves);
19     getLeafSequence(root->right, index, leaves);
20 }
21
22 bool leafSimilar(struct TreeNode* root1, struct TreeNode* root2) {
23     int leaves1[100] = {0};
24     int leaves2[100] = {0};
25
26     int index1 = 0;
27     int index2 = 0;
28
29     getLeafSequence(root1, &index1, leaves1);
30     getLeafSequence(root2, &index2, leaves2);
31
32     if (index1 != index2) {
33         return false;
34     }
35
36     for (int i = 0; i < index1; i++) {
37         if (leaves1[i] != leaves2[i]) {
38             return false;
39         }
40     }
41 }
  
```

Saved to local

Ln 26, Col 20

Accepted Runtime: 0 ms

• Case 1

• Case 2

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
```

Expected

```
true
```

Accepted Runtime: 0 ms

• Case 1

• Case 2

Input

```
root1 =  
[1,2,3]
```

```
root2 =  
[1,3,2]
```

Output

```
false
```

Expected

```
false
```