75 Days of Code Day 34 Problem 104. 872. Leaf-Similar Trees

Type :BST / dfs

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.

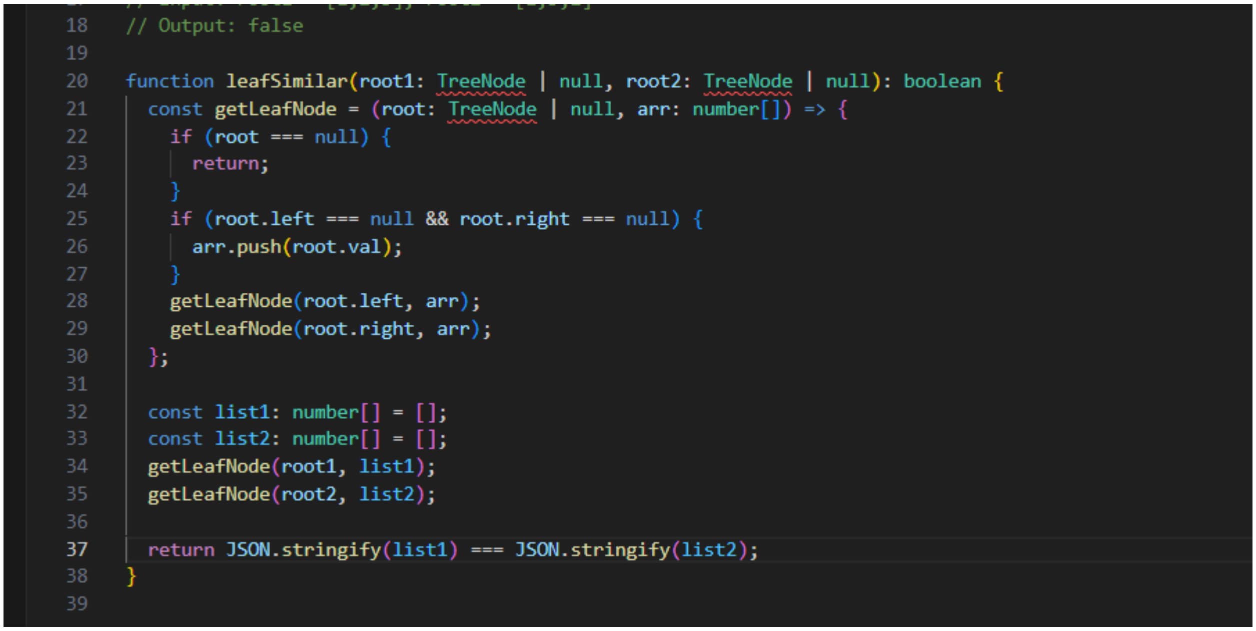
Example 1:

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
Example 2:

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

Solution using DFS

1. Traverse the list with recursion

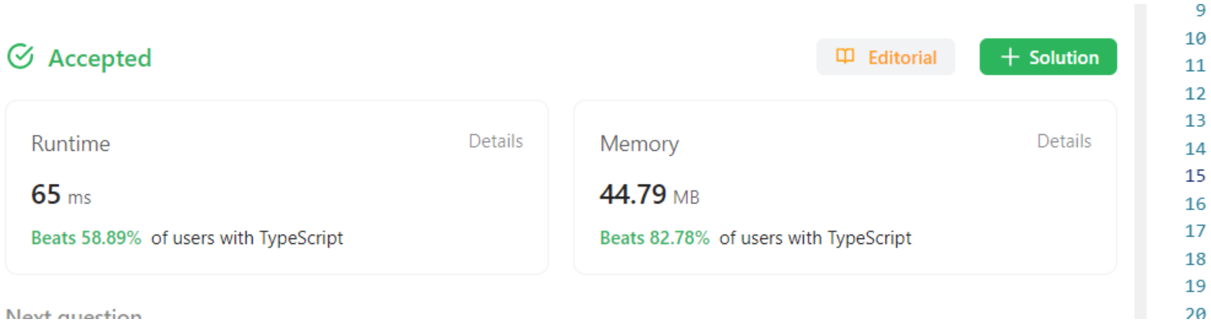


```typescript
// Output: false

function leafSimilar(root1: TreeNode | null, root2: TreeNode | null): boolean {
  const getLeafNode = (root: TreeNode | null, arr: number[]) => {
    if (root === null) {
      return;
    }
    if (root.left === null && root.right === null) {
      arr.push(root.val);
    }
    getLeafNode(root.left, arr);
    getLeafNode(root.right, arr);
  };

  const list1: number[] = [];
  const list2: number[] = [];
  getLeafNode(root1, list1);
  getLeafNode(root2, list2);

  return JSON.stringify(list1) === JSON.stringify(list2);
}
```



Accepted

Runtime Details

65 ms

Beats 58.89% of users with TypeScript

Memory Details

44.79 MB

Beats 82.78% of users with TypeScript

Shubham Agrahari

Shubham Agrahari