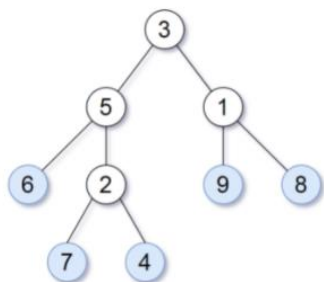


叶子相似的树

请考虑一棵二叉树上所有的叶子，这些叶子的值按从左到右的顺序排列形成一个 **叶值序列**。

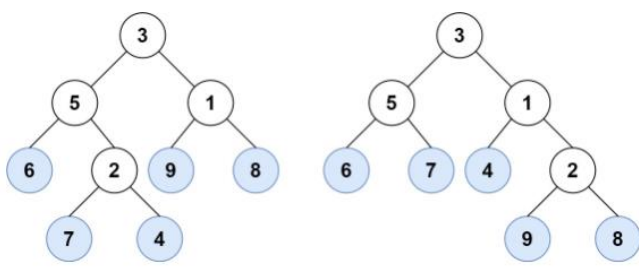


举个例子，如上图所示，给定一棵叶值序列为 (6, 7, 4, 9, 8) 的树。

如果有两棵二叉树的叶值序列是相同，那么我们就认为它们是 *叶相似* 的。

如果给定的两个根结点分别为 root1 和 root2 的树是叶相似的，则返回 true；否则返回 false。

示例 1:



输入: 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]

输出: true

示例 2:

输入: root1 = [1], root2 = [1]

输出: true

示例 3:

输入: root1 = [1], root2 = [2]

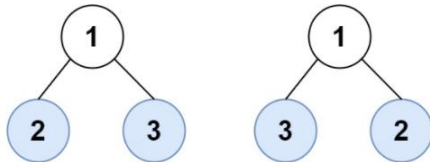
输出: false

示例 4:

输入: root1 = [1,2], root2 = [2,2]

输出: true

示例 5:



输入: root1 = [1,2,3], root2 = [1,3,2]

输出: false

```
# 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:
        res1=[]
        res2=[]
        self.dfs(root1,res1)
        self.dfs(root2,res2)
        return res1==res2

    def dfs(self,root:Optional[TreeNode],result:[list]):
        if not root:
            return
        if not root.left and not root.right:
            result.append(root.val)
        self.dfs(root.left,result)
        self.dfs(root.right,result)
```