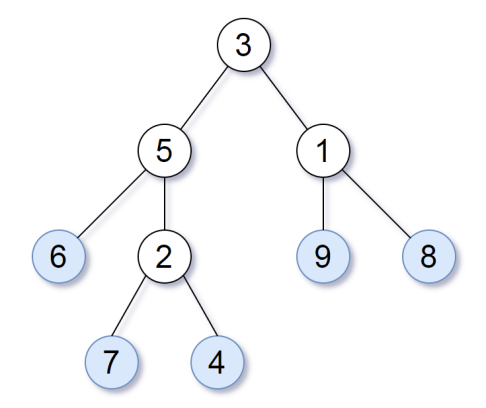
# 题目

请考虑一颗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，给定一颗叶值序列为 (6, 7, 4, 9, 8) 的树。

如果有两颗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。

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

提示：

给定的两颗树可能会有 1 到 200 个结点。

给定的两颗树上的值介于 0 到 200 之间。

# 分析

## 方法一：递归法

/\*\*

\* Definition for a binary tree node.

\* struct TreeNode {

\* int val;

\* TreeNode \*left;

\* TreeNode \*right;

\* TreeNode(int x) : val(x), left(NULL), right(NULL) {}

\* };

\*/

class Solution {

public:

bool leafSimilar(TreeNode\* root1, TreeNode\* root2) {

vector<int> leaves1;

vector<int> leaves2;

dfs(root1, leaves1);

dfs(root2, leaves2);

return leaves1 == leaves2;

}

void dfs(TreeNode\* node, vector<int>& leaves) {

if (node == NULL) return;

if (node->left == NULL && node->right == NULL)

leaves.push\_back(node->val);

dfs(node->left, leaves);

dfs(node->right, leaves);

}

};