

```

1  /*
2  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 。
3
4      3
5     / \
6    5   1
7   / \ / \
8  6  2 9  8
9   / \
10  7  4
11
12 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。
13 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。
14 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
15 false 。
16 提示：
17 给定的两颗树可能会有 1 到 100 个结点。
18 在真实的面试中遇到过这道题？
19 来源：力扣 (LeetCode)
20 链接：https://leetcode-cn.com/problems/leaf-similar-trees
21 著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
22 */

```

分析:

- 利用中序遍历分别两棵树的叶值序列对比即可.

方法一:C++_中序遍历

```

1  /**
2   * Definition for a binary tree node.
3   * struct TreeNode {
4   *     int val;
5   *     TreeNode *left;
6   *     TreeNode *right;
7   *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
8   * };
9   */
10 class Solution
11 {
12
13     private:
14
15     void genSeriesRoot ( vector<int>& vi ,
16                         TreeNode* node
17                         )
18     {
19         if(node==NULL)
20         {
21             return;
22         }
23
24         genSeriesRoot(vi,node->left);

```

```

26         if(node->left==NULL && node->right== NULL)
27         {
28             vi.push_back(node->val);
29             return;
30         }
31         genSeriesRoot(vi,node->right);
32     }
33
34
35     public:
36     bool leafSimilar(TreeNode* root1, TreeNode* root2)
37     {
38
39         vector<int>    vi1;
40         vector<int>    vi2;
41         genSeriesRoot(vi1,root1);
42         genSeriesRoot(vi2,root2);
43
44         if(vi1.size()!=vi2.size())
45         {
46             return false;
47         }
48
49         for(int i = 0 ; i < vi1.size() ;i++)
50         {
51             if(vi1[i]!=vi2[i])
52             {
53                 return false;
54             }
55         }
56         return true;
57     }
58 };
59
60 /*
61 执行结果:
62 通过
63 显示详情
64 执行用时 :8 ms, 在所有 cpp 提交中击败了58.47% 的用户
65 内存消耗 :14.1 MB, 在所有 cpp 提交中击败了64.13%的用户
66 */

```