

[每日 LeetCode] 100. Same Tree

作者: [Hanseltu](#)

原文链接: <https://ld246.com/article/1558453607392>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

原文链接 [\[每日LeetCode\] 100. Same Tree](#)

Description:

Given two binary trees, write a function to check if they are the same or not.

Two binary trees are considered the same if they are structurally identical and the nodes have the same value.

Example 1:

Input:

```
  1      1
 / \    / \
2   3  2   3

[1,2,3], [1,2,3]
```

Output: true

Example 2:

Input:

```
  1      1
 /      \
2        2

[1,2], [1,null,2]
```

Output: false

Example 3:

Input:

```
  1      1
 / \    / \
2   1  1   2

[1,2,1], [1,1,2]
```

Output: false

思路：本题判断两个树是否是同一棵，同样可以采用递归和非递归方法实现。递归方法比较简洁，分对当前结点，左结点和右结点依次比较即可。非递归可采用层次、先序、中序和后序遍历的任何一种这里使用中序遍历的非递归写法，遍历后判断相应位置上的结点是否相同即可。

C++代码（递归方法）

```
/**
 * 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 isSameTree(TreeNode* p, TreeNode* q) {
        if (p == NULL || q == NULL) return (p == q);
        return (p->val == q->val && isSameTree(p->left, q->left) && isSameTree(p->right, q->right));
    }
};

```

运行时间: 4ms

运行内存: 10M

C++代码 (非递归方法)

```

/**
 * 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 isSameTree(TreeNode* p, TreeNode* q) {
        stack<TreeNode*> pcallStack;
        stack<TreeNode*> qcallStack;

        TreeNode* pit = p;
        TreeNode* qit = q;

        if(!pit && qit || (pit && !qit))
            return false;

        while((pit && qit) || (!pcallStack.empty() && !qcallStack.empty())) {
            if(!pit && qit || (pit && !qit))
                return false;

            if(pit && qit) {
                pcallStack.push(pit);
                pit=pit->left;

                qcallStack.push(qit);
            }
        }
    }
};

```

```
        qit=qit->left;
    } else {
        if(pcallStack.top()->val != qcallStack.top()->val)
            return false;

        pit = pcallStack.top()->right;
        pcallStack.pop();
        qit = qcallStack.top()->right;
        qcallStack.pop();
    }
}

return true;
};
```

运行时间: 4ms

运行内存: 10.1M