

100 Same Tree

2018年3月30日 15:53

Question:

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

来自 <<https://leetcode.com/problems/same-tree/description/>>

给定两个二叉树，写一个函数来检查它们是否相同。

如果两棵树在结构上相同并且节点具有相同的值，则认为它们是相同的。

Solution for Python3:

```
1 # Definition for a binary tree node.
2 # class TreeNode:
3 #     def __init__(self, x):
4 #         self.val = x
5 #         self.left = None
6 #         self.right = None
7
8 class Solution1:
9     def isSameTree(self, p, q):
10         """
11         :type p: TreeNode
12         :type q: TreeNode
13         :rtype: bool
14         """
15         if p and q:
16             return p.val == q.val and self.isSameTree(p.left, q.left) and self.isSameTree(p.right, q.right)
17         return p == q
18
19 class Solution2:
20     def isSameTree(self, p, q):
21         """
22         :type p: TreeNode
23         :type q: TreeNode
24         :rtype: bool
25         """
26         return p == q if (not p or not q) else p.val == q.val and self.isSameTree(p.left, q.left) and self.isSameTree(p.right, q.right)
```

Solution for C++:

```
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 public:
12     bool isSameTree(TreeNode* p, TreeNode* q) {
13         return (p == NULL || q == NULL) ? (p == q) : (p->val == q->val) && isSameTree(p->left, q->left) && isSameTree(p->right, q->right);
14     }
15 };

```