

剑指Offer（五十八）：对称的二叉树

🕒 2018年1月26日 11:02:40 📄 4 🌡 4,590 °C 🛠 编辑



一、前言

本系列文章为《剑指Offer》刷题笔记。

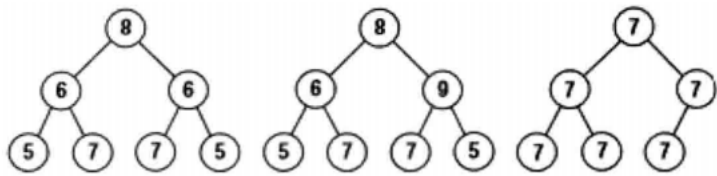
刷题平台：[牛客网](#)

书籍下载：[共享资源](#)

二、题目

请实现一个函数，用来判断一颗二叉树是不是对称的。注意，如果一个二叉树同此二叉树的镜像是同样的，定义其为对称的。

1、思路



我们通常有三种不同的二叉树遍历算法，即前序遍历、中序遍历和后序遍历。在这三种遍历算法中，都是先遍历左子结点再遍历右子结点。以前序遍历为例，我们可以定义一个遍历算法，先遍历右子结点再遍历左子结点，暂且称其为前序遍历的对称遍历。

遍历第一棵树，前序遍历的遍历序列为{8,6,5,7,6,7,5}，其对称遍历的遍历序列为{8,6,5,7,6,7,5}。

遍历第二棵树，前序遍历的遍历序列为{8,6,5,7,9,7,5}，其对称遍历的遍历序列为{8,9,5,7,6,7,5}。

可以看到，使用此方法可以区分前两棵树，第一棵树为对称树，第二棵树不是对称树。但是当使用此方法，你会发现第三棵树的前序遍历和对称前序遍历的遍历序列是一样的。

怎么区分第三棵树呢？解决办法就是我们也要考虑NULL指针。此时，前序遍历的遍历序列{7,7,7,NULL,NULL,7,NULL,NULL,7,7,NLL,NULL,NULL}，其对称遍历的遍历序列为{7,7,NULL,7,NULL,NULL,7,7,NULL,NULL,7,NULL,NULL}。因为两种遍历的序列不同，因此这棵树不是对称树。

2、代码

C++:

```
1  /*
2  struct TreeNode {
3      int val;
4      struct TreeNode *left;
5      struct TreeNode *right;
6      TreeNode(int x) :
7          val(x), left(NULL), right(NULL) {
8      }
9  };
10 */
11 class Solution {
12 public:
13     bool isSymmetrical(TreeNode* pRoot)
14     {
15         if(pRoot == NULL){
```

```
16         return true;
17     }
18     return isSymmetricalCor(pRoot, pRoot);
19 }
20 private:
21 bool isSymmetricalCor(TreeNode* pRoot1, TreeNode* pRoot2){
22     if(pRoot1 == NULL && pRoot2 == NULL){
23         return true;
24     }
25     if(pRoot1 == NULL || pRoot2 == NULL){
26         return false;
27     }
28     if(pRoot1->val != pRoot2->val){
29         return false;
30     }
31     return isSymmetricalCor(pRoot1->left, pRoot2->right) && isSymmetricalCor(pRoot1->right, pRoot2->left);
32 }
33 };
```

Python:

感谢@小小毛提供python代码记本地测试用例。

	Pytho
<pre>1  #- coding:utf-8 -*- 2  class TreeNode: 3      def __init__(self, x): 4          self.val = x 5          self.left = None 6          self.right = None 7  class Solution: 8      def isSymmetrical(self, pRoot): 9          # write code here 10         if not pRoot: 11             return True 12         return self.recursiveTree(pRoot.left, pRoot.right) 13 14     def recursiveTree(self, left, right): 15         if not left and not right: 16             return True 17         if not left or not right: 18             return False 19         if left.val == right.val: 20             return self.recursiveTree(left.left, right.right) and self.recursiveTree(left.right, right.left) 21         return False 22 if __name__ == '__main__': 23 24     A1 = TreeNode(1) 25     A2 = TreeNode(2) 26     A3 = TreeNode(2) 27     A4 = TreeNode(3) 28     A5 = TreeNode(4) 29     A6 = TreeNode(4) 30     A7 = TreeNode(3) 31 32     A1.left=A2 33     A1.right=A3 34     A2.left=A4 35     A2.right=A5 36     A3.left=A6 37     A3.right=A7 38 39 40     solution = Solution() 41     ans=solution.isSymmetrical(A1) 42     print(ans)</pre>	



微信公众号

分享技术，乐享生活：微信公众号搜索

「JackCui-AI」关注一个在互联网摸爬滚

打的潜行者。

知之者不如好之者，好之者不如乐之者。--- 孔子