

剑指Offer（十七）：树的子结构

🕒 2017年12月8日 15:07:12 🗂 8 🌡 5,639 °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

输入两颗二叉树A，B，判断B是不是A的子结构。（PS：我们约定空树不是任意一个树的子结构）。

1、思路

要查找树A中是否存在和树B结构一样的子树，我们可以分为两步：第一步在树A中找到和B的根结点的值一样的结点R，第二步再判断树A中以R为根结点的子树是不是包含和树B一样的结构。

这里使用递归的方法即可。

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 class Solution {
11 public:
12     bool HasSubtree(TreeNode* pRoot1, TreeNode* pRoot2)
13     {
14         bool result = false;
15         if(pRoot1 != NULL && pRoot2 != NULL){
16             if(pRoot1->val == pRoot2->val){
17                 result = DoesTree1HasTree2(pRoot1, pRoot2);
18             }
19             if(!result){
20                 result = HasSubtree(pRoot1->left, pRoot2);
21             }
22             if(!result){
23                 result = HasSubtree(pRoot1->right, pRoot2);
24             }
25         }
26         return result;
27     }
28 private:
29     bool DoesTree1HasTree2(TreeNode* pRoot1, TreeNode* pRoot2){
30         if(pRoot2 == NULL){
31             return true;
32         }
33         if(pRoot1 == NULL){
34             return false;
35         }
36         if(pRoot1->val != pRoot2->val){
37             return false;
```

```
38         }
39         return DoesTree1HasTree2(pRoot1->left, pRoot2->left) && DoesTree1HasTree2(pRoot1->right, pRoot2->right);
40     }
41 };
```

Python2.7:

<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 HasSubtree(self, pRoot1, pRoot2): 9 # write code here 10 if not pRoot1 or not pRoot2: 11 return False 12 return self.HasSubtree(pRoot1.left, pRoot2) or self.HasSubtree(pRoot1.right, pRoot2) or self.is_subtree(pRoot1, pRoot2) 13 def is_subtree(self, A, B): 14 if not B: 15 return True 16 if not A or A.val != B.val: 17 return False 18 return self.is_subtree(A.left, B.left) and self.is_subtree(A.right, B.right)</pre>	Pytho
--	-------



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。

悲剧将人生的有价值的东西毁灭给人看，喜剧将那无价值的撕破给人看。--- 鲁迅