

剑指Offer（二十四）：二叉树中和为某一值的路径

🕒 2017年12月12日 11:38:42 📄 15 🌡 5,143 °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

输入一颗二叉树和一个整数，打印出二叉树中结点值的和为输入整数的所有路径。路径定义为从树的根结点开始往下一直到叶结点所经过的结点形成一条路径。

1、思路

深度优先搜索。使用前序遍历，使用两个全局变量result和tmp，result来存放最终结果，tmp用来存放临时结果。

每次遍历，我们先把root的值压入tmp，然后判断当前root是否同时满足：

- 与给定数值相减为0；
- 左子树为空；
- 右子树为空。

如果满足条件，就将tmp压入result中，否则，依次遍历左右子树。需要注意的是，遍历左右子树的时候，全局变量tmp是不清空的，直到到了根结点才清空tmp。

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     vector<vector<int> > FindPath(TreeNode* root,int expectNumber){
13         if(root == NULL){
14             return result;
15         }
16
17         tmp.push_back(root->val);
18         if((expectNumber - root->val) == 0 && root->left == NULL && root->right == NULL){
19             result.push_back(tmp);
20         }
21
22         //遍历左子树
23         FindPath(root->left, expectNumber - root->val);
24         //遍历右子树
25         FindPath(root->right, expectNumber - root->val);
26     }
```


```
27         tmp.pop_back();
28         return result;
29     }
30 private:
31     vector<vector<int> > result;
32     vector<int> tmp;
33 };
```

Python：

<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     # 返回二维列表，内部每个列表表示找到的路径 9     def FindPath(self, root, expectNumber): 10         # write code here 11         if not root: 12             return [] 13         if not root.left and not root.right and expectNumber == root.val: 14             return [[root.val]] 15         res = [] 16         left = self.FindPath(root.left, expectNumber-root.val) 17         right = self.FindPath(root.right, expectNumber-root.val) 18         for i in left+right: 19             res.append([root.val]+i) 20         return res</pre>	Pytho
---	-------

感谢@小小毛的本地测试用例：

<pre>1 class TreeNode: 2     def __init__(self, x): 3         self.val = x 4         self.left = None 5         self.right = None 6 7 class Solution: 8     # 返回二维列表，内部每个列表表示找到的路径 9     def FindPath(self, root, expectNumber): 10         # write code here 11         if not root: 12             return [] 13         if not root.left and not root.right and expectNumber == root.val: 14             return [[root.val]] 15         res = [] 16         left = self.FindPath(root.left, expectNumber-root.val) 17         right = self.FindPath(root.right, expectNumber-root.val) 18         for i in left+right: 19             res.append([root.val]+i) 20         return res 21 if __name__=='__main__': 22     A1 = TreeNode(1) 23     A2 = TreeNode(2) 24     A3 = TreeNode(3) 25     A4 = TreeNode(4) 26     A5 = TreeNode(5) 27     A6 = TreeNode(6) 28 29     A1.left=A2 30     A1.right=A3 31     A2.left=A4 32     A2.right=A5 33     A4.left=A6 34 35     solution=Solution() 36     ans=solution.FindPath(A1,8) 37     print('ans=',ans)</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。

友谊是两颗心真诚相待，而不是一颗心对另一颗心的敲打。--- 鲁迅