

剑指Offer（十八）：二叉树的镜像

🕒 2017年12月9日 10:06:54 📄 6 🌡 5,316 °C 🛠 编辑



一、前言

本系列文章为《剑指Offer》刷题笔记。

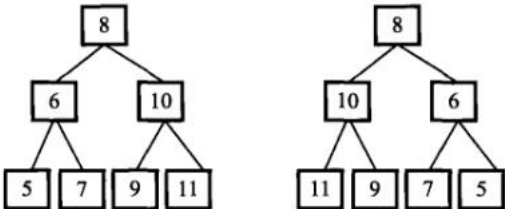
刷题平台：[牛客网](#)

书籍下载：[共享资源](#)

二、题目

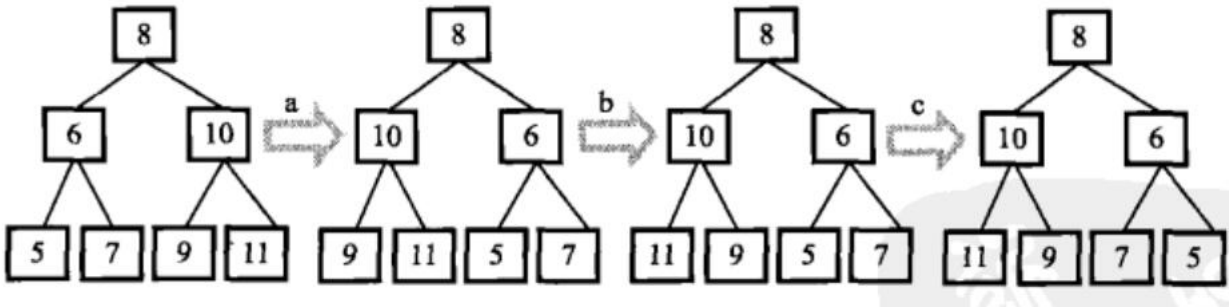
操作给定的二叉树，将其变换为源二叉树的镜像。

如下图所示：



1、思路

先交换根节点的两个子结点之后，我们注意到值为10、6的结点的子结点仍然保持不变，因此我们还需要交换这两个结点的左右子结点。做完这两次交换之后，我们已经遍历完所有的非叶结点。此时变换之后的树刚好就是原始树的镜像。交换示意图如下所示：



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  }
```

```
9  };*/
10 class Solution {
11 public:
12     void Mirror(TreeNode *pRoot) {
13         if((pRoot == NULL) || (pRoot->left == NULL && pRoot->right == NULL)){
14             return;
15         }
16
17         //交换根节点的左右结点
18         TreeNode *pTemp = pRoot->left;
19         pRoot->left = pRoot->right;
20         pRoot->right = pTemp;
21
22         //递归左子树
23         if(pRoot->left){
24             Mirror(pRoot->left);
25         }
26         //递归右子树
27         if(pRoot->right){
28             Mirror(pRoot->right);
29         }
30     }
31 };
```

Python2.7:

```
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  # 返回镜像树的根节点
9  def Mirror(self, root):
10     # write code here
11     if (root == None or (root.left == None and root.right == None)):
12         return None
13     tmp = root.left
14     root.left = root.right
15     root.right = tmp
16
17     if root.left:
18         self.Mirror(root.left)
19     if root.right:
20         self.Mirror(root.right)
```

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。

一个人思虑太多，就会失去做人的乐趣。--- 莎士比亚