

第3题——二叉树的镜像

请完成一个函数，输入一个二叉树，该函数输出它的镜像。

例如输入：



镜像输出：



示例 1:

输入: root = [4,2,7,1,3,6,9]

输出: [4,7,2,9,6,3,1]

限制: $0 \leq \text{节点个数} \leq 1000$

前序递归解法（1）

第一种解法，使用递归（前序遍历），让每一个节点的左右孩子互换位置

```

struct TreeNode* mirrorTree(struct TreeNode* root){
    if(root)
    {
        struct TreeNode* temp=root->left;
        root->left=root->right;
        root->right=temp;
        mirrorTree(root->left);
        mirrorTree(root->right);
    }
    return root;
}
  
```

需要注意的是，这种方法必须要检查传入的二叉树是否合法，即该树不为空，否则无法执行。

前序递归简洁版（2）

```

struct TreeNode* mirrorTree(struct TreeNode* root){

    if(root)
  
```

```

{
    struct TreeNode *temp=root->left;
    root->left=mirrorTree(root->right);
    root->right=mirrorTree(temp);
}
return root;
}

```

很厉害，代码简洁，而且可读性强

C++可以使用完全相同的办法

```

class Solution {
public:
    TreeNode* mirrorTree(TreeNode* root) {
        if(root)
        {
            TreeNode *temp=root->left;
            root->left=mirrorTree(root->right);
            root->right=mirrorTree(temp);
        }
        return root;
    }
};

```