

2. Write a C++ function, *SwapTree()*, that swaps the left and right children of every node of a binary tree. An example is given in Figure 5.19.

题目说明：

翻转二叉树，不止根节点的左右子树需要交换，树中的每一个节点的左右子树都需要交换。

解法：递归

leetcode相关题目：[226. Invert Binary Tree](#)

5. Write an algorithm to construct the binary tree with given preorder and inorder sequences.

题目说明：

根据一棵树的前序遍历与中序遍历构造二叉树。

例子：

前序遍历 preorder = [3,9,20,15,7]

中序遍历 inorder = [9,3,15,20,7]

解法：

1. 前序遍历的形式：

[根节点, [左子树的前序遍历结果], [右子树的前序遍历结果]]

2. 中序遍历的形式：

[[左子树的中序遍历结果], 根节点, [右子树的中序遍历结果]]

3. 使用**前序遍历**中给出的**根节点**，在**中序遍历**中找到**根节点对应的位置**，将**中序遍历**分为**两部分**，则可以得到**左子树和右子树的大小**，由此可以在前序遍历中找到左子树和右子树的前序遍历的序列。

4. 由第三步能得到左子树和右子树的前序遍历和中序遍历的序列，问题转换为

根据前序遍历和中序遍历构建左子树和右子树的子问题
用递归求解。

*leetcode*相关题目: [105. Construct Binary Tree from Preorder and Inorder Traversal](#)

```
1 //Definition for a binary tree node.
2 struct TreeNode {
3     int val;
4     TreeNode *left;
5     TreeNode *right;
6     TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
7 };
8
9 void printTree(TreeNode* node) {
10
11 }
12
13 int main() {
14     TreeNode * node1 = new TreeNode(1);
15     TreeNode * node2 = new TreeNode(2);
16     TreeNode * node3 = new TreeNode(3);
17     TreeNode * node4 = new TreeNode(4);
18     TreeNode * node5 = new TreeNode(5);
19
20     node1->left = node2;
21     node1->right = node3;
22     node2->left = node4;
23     node3->right = node5;
24 }
```