

## 简单问题可以不简单

### 226.翻转二叉树

翻转一棵二叉树。

输入：



输出：



只要根据层次遍历算法,每次访问一个结点就将该结点的左右子树换一个顺序  
或者是通过先序遍历算法或者后序遍历算法把左右子树调换

```
1 class Solution {
2 public:
3     TreeNode* invertTree(TreeNode* root) {
4         queue<TreeNode*> que;
5         if(root==NULL) return root;
```

```
6     que.push(root);
7     while(!que.empty()){
8         int size=que.size();
9         TreeNode* node=que.front();
10        que.pop();
11        if(node->right&&node->left){
12            TreeNode* tmp=node->right;
13            node->right=node->left;
14            node->left=tmp;
15            que.push(node->left);
16            que.push(node->right);
17            continue;
18        }
19        if(node->left){
20            que.push(node->left);
21            node->right=node->left;
22            node->left=NULL;
23            continue;
24        }
25        if(node->right){
26            que.push(node->right);
27            node->left=node->right;
28            node->right=NULL;
29            continue;
30        }
31    }
32    return root;
33 }
34 };
```