二叉树的修改和构造

翻转二叉树

思路：利用**层序遍历**的程序框架，每次交换队列首节点的左节点和右节点（实际是把左边整体和右边整体交换），交换函数在java需要自行编写然后在主函数中调用

题目：

226. 翻转二叉树（简单）

从前序与中序遍历序列构造二叉树

属于**二叉树的构造问题**，可以采用递归，也可以采用迭代，本题采用**栈的迭代**，思路详情看leetcode官方解答

“从中序与后序遍历序列构造二叉树”与之类似，区别在于遍历时是从后往前，注意对比两者代码不同

题目：

106. 从中序与后序遍历序列构造二叉树（中等）

105. 从前序与中序遍历序列构造二叉树（中等）

最大二叉树

最大二叉树：对二叉树的每个节点而言，它的子节点要么为空，要么全比它自己小

此题同属于**二叉树的构造问题**，采用**递归**方法解决该题

这题还要用到**”切割“的思想**，每次要找到递归函数处理数组里面的最大值，并确定最大值在数组里的位置，以此做分割

递归三部曲：

（1）递归参数：首先要传入数组，其次是分割的左边界和右边界；返回类型为节点

（2）终止条件：当左边界大于右边界时，返回空

（3）递归逻辑：分别对当前节点的左节点和右节点使用递归函数（以处理数组最大值为界）

题目：

112. 最大二叉树（中等）

合并二叉树

采取**层序遍历**的方式，利用“**两两打包**”的思想进行二叉树合并

如果两棵树均为空，则返回null；如果root1为空，则返回root2，反之亦然

当两棵树根节点均不为空时，分为**以下2种情况**：

（1）队列首两个节点node1，node2的左/右节点分别都不为空，则将它们对应的左/右节点加入队列，且执行正常操作，把它们的和赋给node1

（2）队列首两个节点node1，node2当中node1不含左/右节点，则直接把node2对应节点赋给node1

隐含情况（不进行操作）：

（1）node1含有左/右节点而node2不含有，则对应节点就是原来node1的值不变

（2）node1，node2均不含左/右节点，则对应节点为空

*总结：每次循环加入队列的节点数量都是2/4个（1对或两对），此时对应两棵树对应位置的节点均不为空的情况

题目：

617. 合并二叉树（简单）