

# 第五周实验报告

沈家成

2017 年 10 月 17 日

## 1 1111 二哥学二叉树

### 1.1 问题分析

这是一个递归的问题，难点在于应该往下传递哪些递归参数，才能达到递归解决的效果哦。为了理清整个过程，需要首先手工模拟这个过程。

### 1.2 过程分析

第一步，找到前序遍历的第一个元素，作为根节点。将根节点放入顺序数组。

第二步，找到根节点在中序遍历中的位置，从而将中序遍历分为左右两部分。

第三步，找到中序遍历左边/右边部分对应的前序遍历，将当前根节点在顺序数组中的位置，左/右半中序遍历，及其前序遍历传入递归函数。

### 1.3 代码细节

递归终止的条件：当新的前序排列只剩一个元素的时候，只要将其放入顺序数组，不用再往下递归了。

如何查找中序遍历元素对应的前序元素：经过观察，发现左/右半中序遍历总是在前序中成团出现，因此只要找到任意左/右半中序元素在前序遍历中的位置，就找到了整个左/右半中序遍历对应的前序遍历元素。

## 2 1211 isCBT

### 2.1 难点分析

从输入的数据获得二叉树的结构不是很难，困难的在于如何判断是不是完全二叉树。完全二叉树只允许最后一层不满，并且叶子结点都要靠左。叶子结点容易分辨，但是怎么知道它处于最底层哪个位置呢？就要往上搜索，一直回溯到根结点才可以进行判断，这样就很复杂。

### 2.2 解决方法

顺序数组存储二叉树的时候，如果是完全二叉树，元素中间是不会有空洞的，因此可以利用这个性质，判断是否为二叉树。

首先寻找根结点。通过搜寻没有成为子结点的结点来寻找根结点。然后从根结点开始遍历，将结点编号储存进顺序二叉树数组。最后判断数组中是否有空洞，从而判断

## 2.3 再思考

在进行数组顺序储存的时候，常常会越界。一种解决办法是算出层数，预留空间；另一种办法是每次储存时进行判断是否越界，后来进一步想，为什么会越界呢？因为不够存储。那又为什么不够存储？说明之前有空余的空间，正好就是因为不是完全二叉树。因此，只要判断储存数组时会不会越界，就可以间接判断是不是完全二叉树了。

## 3 1214 traverse

### 3.1 孩子兄弟表示法

孩子兄弟表示法的前序遍历和后序遍历只需要用递归的方法，重复调用自身就可以了。然而，层次遍历因为要分层输出，就不适用于递归了，而是适用队列的方法。

### 3.2 具体实现

首先输出根结点，并将根结点的孩子们依次放入队列，这样下一步的任务就是层次遍历这些孩子结点。然后，从队列读取任务，输出当前结点，并将当前结点的孩子们加入队列任务，为下一层结点输出做准备。这样一直重复，直到队列为空，就利用队列后进后出的特性，实现了队列。

### 3.3 小结

根据实际情况的需求，选择数据结构，会极大地方便思路的整理和代码的实现。