

实验 3 日志

一. 实验设计

实验三树的实现要求我们基于左子节点/右子节点表示法或者二叉链表来实现二叉树的 ADT，首先明确树的 ADT 的数据对象和基本操作，我们可以对树的结点 ADT 进行实现或者对树的 ADT 进行实现，结点 ADT 数据对象包括树结点存放的值，左子树指针，右子树指针，基本操作主要有设置左右结点，得到结点的值，叶子结点的判断等等，树的 ADT 数据对象则包括根结点，左右子树，基本操作主要是对于结点的增删，树结构的遍历，前序，中序，后序，层次遍历等。题目本身要求我们根据中序，后序遍历，构建出整颗树，然后通过层次遍历输出树层次遍历的各个结点的值。

整体代码框架就是实现二叉树 ADT，定义一个二叉树类，继承抽象树类，结点类单独实现，在抽象树类和具体实现的树类中创建结点对象，最后完成构建树，从而实现树的各种遍历。

二. 实验记录

2019 年 10 月 20 日

如何对二叉树进行遍历：起初的想法是想一级一级遍历下去，但当树的规模增大后，似乎有点不太思想出这个过程。课本上通过递归的方式进行遍历，这样的话就将整个问题分解，化简为简单的容易处理的小问题，主要为三个部分，根结点，对左右子树的递归遍历。

2019 年 10 月 23 日

主程序测试的编写：利用两个数组输入二叉树的中序遍历和后序

遍历，关键步骤在于如何利用这两个序列构建出整个二叉树，思路就是根据根结点的位置确定出中序中左右子树，然后左右子树利用递归的想法和上面一样确定出。在写实验代码的过程中，出现的难点在于如何去找到每次递归的一个子树的中序序列，这里我采用的是数组下标之间的关系来确定每次的中序序列，用后序的根结点确定。

二. 心得和问题

2019 年 10 月 22-23 日

1. 在写实验的时候，查了资料，对于树的遍历，采用递归的方式进行处理，实际上是分治思想的应用，将问题分解为相似的小问题，使问题更容易处理。树的遍历和构建用的就是这种思想，包括对根结点和左右子树的处理，最简单的问题形式就是处理根结点或者一棵空树。
2. 指针使用上的问题，用链表来处理问题时，要注意指针的初始化，不然在后续处理上可能对某些操作存在影响，造成程序异常出错。