

哈爾濱工業大學

# 数据结构实验报告

## 实验二

专        业    计算机类

学        号    120L022314

学        生    瞿久尧

指 导 教 师    苗东菁老师

## 一. 简介/问题求解思路

1. 问题：给出一个树的前序遍历和中序遍历结果，构造二叉树，并输出层序遍历结果。
2. 求解：由于给出了前序和中序遍历结果，则二叉树是唯一的，只需要根据前序和中序的特点构造二叉树即可，利用前序找到树的根结点，在中序中找到此结点就可以把树中序分为左子树、根、右子树三个部分，再分别对左子树和右子树重复以上操作就可以建立起一棵树，最后再用队列对树进行层序遍历。

## 二. 数据结构设计与实现

1. 逻辑结构：二叉树（建树）、栈（非递归）、队列（层序遍历）
2. 存储结构：二叉链表、数组
3. 实现
  - 3.1 根据前序和中序遍历结果，对左子树和右子树反复操作构建二叉树，根结点有两个指针 ltree、rtree 分别指向左子树和右子树的根结点，可利用递归或非递归方式对左右子树进行操作（非递归方法需要手动建栈）。
  - 3.2 二叉树建好后则创建长度为 n 的数组，并形成队列，从头指针开始入队，再入队左孩子和右孩子，当孩子结点都入队后，即可出队并输出，形成层序遍历。

## 三. 算法设计及分析

1. 函数设计
  - 1.1 Creat() 函数，创建数组存储输入的前序和中序遍历结果，并且建立起二叉树的根结点。
  - 1.2 MakeNULL() 函数，判断树是否为空，并且可在程序末尾对使用的链表进行释放。
  - 1.3 LTree() 函数，导入左子树在中序遍历中的范围，以及需要建立的二叉树，在这个范围中利用前序遍历找到根结点，进行下一步递归。LTree() 中同时包含 LTree() 和 RTree()，进行递归。
  - 1.4 RTree() 函数，导入右子树在中序遍历中的范围，以及需要建立的二叉树，在这个范围中利用前序遍历找到根结点，进行下一步递归。RTree() 中同时包含 LTree() 和 RTree()，进行递归。

1.5 Ceng()函数，建立队列，对树进行层序遍历。

1.6 Fei()函数，进行非递归建立二叉树，其中包括建立标志数组，记录某结点的孩子结点访问了几个，并且手动建栈，当某结点已经访问过两个孩子结点，则出栈。不断入栈和弹栈建立左右子树。

## 2. 性能分析

### 2.1 时间复杂度

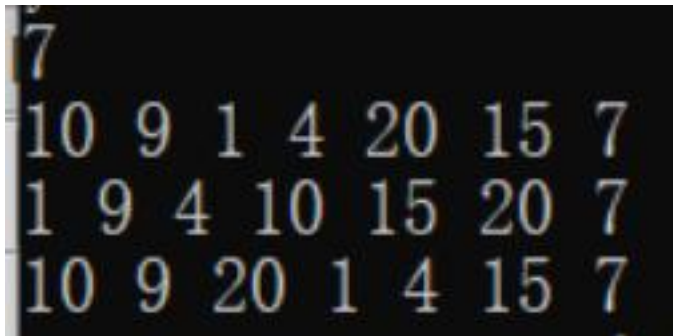
2.1.1 遍历左子树和右子树，由于中序和前序遍历的不同，时间复杂度可能在  $O(n)$  -  $O(n^2)$  不等，平均性能应该为  $O(n^2)$ ；

2.1.2 层序遍历时间复杂度为  $O(n)$ 。

### 2.2 空间复杂度

二叉树空间复杂度为  $O(n)$ ，以及一些其他长度为  $n$  的数组，总的空间复杂度也为  $O(n)$ 。

## 四. 实验测试结果及结果分析



给出结点的个数，然后依次输出前序和中序遍历的结果就可以得到层序遍历的结果

## 五. 实验中遇到的问题及解决方法

问题：非递归遍历时需要手动建栈，但不方便确定何时弹栈

解决：可以建立一个标志数组，数组下标对应结点在前序遍历中的顺序，各元素的值代表访问了此结点的几个孩子，每次经过这个结点时，该结点值+1，若访问了两个孩子代表这个结点可以弹栈，以此来进行规律的入栈和弹栈