|  |
| --- |
| 哈尔滨工业大学(深圳) |
| **《数据结构》实验报告** |
|  |
| 实验3  [键入文档副标题]  学 院: 计算机科学与技术   |  |  | | --- | --- | | 姓 名: | 段佳慧 | | 学 号: | 200110807 | | 专 业: | 计算机科学与技术 | | 日 期: | 2021-04-14 | |

# 问题分析

1. 按层次遍历构建二叉树，并输出该树的前序、中序、后序遍历。即在构建完成二叉树之后，可以通过递归的方式输出该树的三种遍历形式。
2. 采用深度优先遍历，求该二叉树的最大路径和。
3. 求所有的左子叶的权值和。可以通过递归实现，先找到所有的左子叶，最后结果相加。
4. 通过递归求取该树的镜像，即翻转该二叉树。通过递归不断对二叉树的子树进行翻转，最后可将整个二叉树翻转。

# 二、详细设计

## 2.1 设计思想

1、可以用队列来实现树的结构，而队列又可以用栈来实现

2、不断递归调用函数实现对该树的三种遍历输出

3、对于所有左子叶的权值和，运用递归找到所有的左子叶

4、树的镜像即将结点翻转。注意此时树的形态变化

## 2.2 存储结构及操作

(1) 存储结构 存储结构如下：

typedef struct TreeNode{

int id;

int val;

struct TreeNode \*left;

struct TreeNode \*right;

} TreeNode, \*TreeNodePtr;

typedef struct ListNode{

struct TreeNode \*node; // 队列的值的类型是树节点指针

struct ListNode \*next;

} ListNode, \*ListNodePtr;

typedef struct Queue{

ListNodePtr dummyHead;

ListNodePtr tail;

int size;

} \*QueuePtr;

(2) 涉及的操作（一般为自定义函数，可不写过程，但要注明该函数的含义。）

* /\*通过队列来实现层次遍历构建二叉树，并返回二叉树的头结点\*/

TreeNodePtr createTreeWithLevelOrder(int \*data, int size)

* /\*前序遍历\*/

void preOrderTraverse(TreeNodePtr root)

* /\*中序遍历\*/

void inOrderTraverse(TreeNodePtr root)

* /\*后序遍历\*/

void postOrderTraverse(TreeNodePtr root)

* /\*通过深度优先遍历来求取该二叉树的最大路径和\*/

int maxPathSum(TreeNodePtr root, int sum)

* /\*通过递归求取该二叉树的所有左子叶权重之和\*/

int sumOfLeftLeaves(TreeNodePtr root)

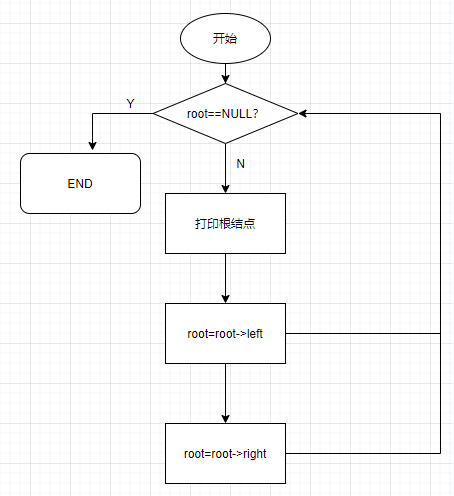
* 通过递归求取该树的镜像，即翻转该二叉树TreeNodePtr invertTree(TreeNodePtr root)

## 2.3 程序整体流程

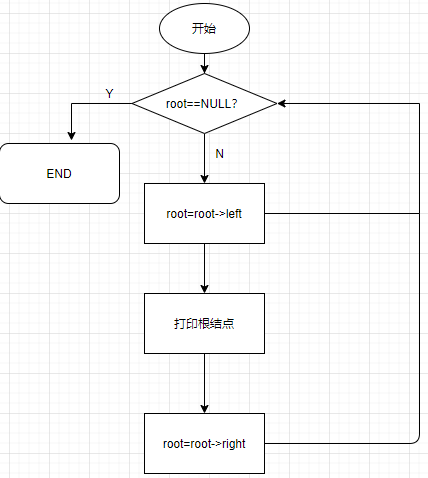
画出整体流程，及核心算法流程。

核心算法流程：

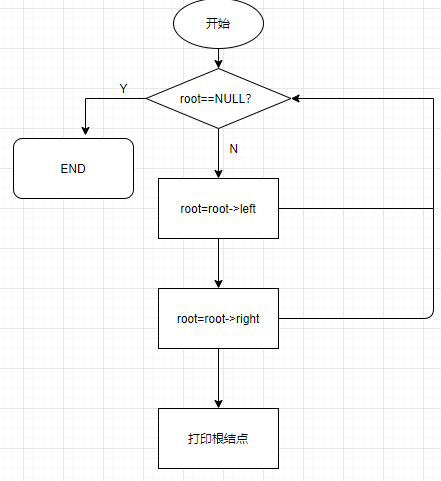
* 递归实现前序遍历



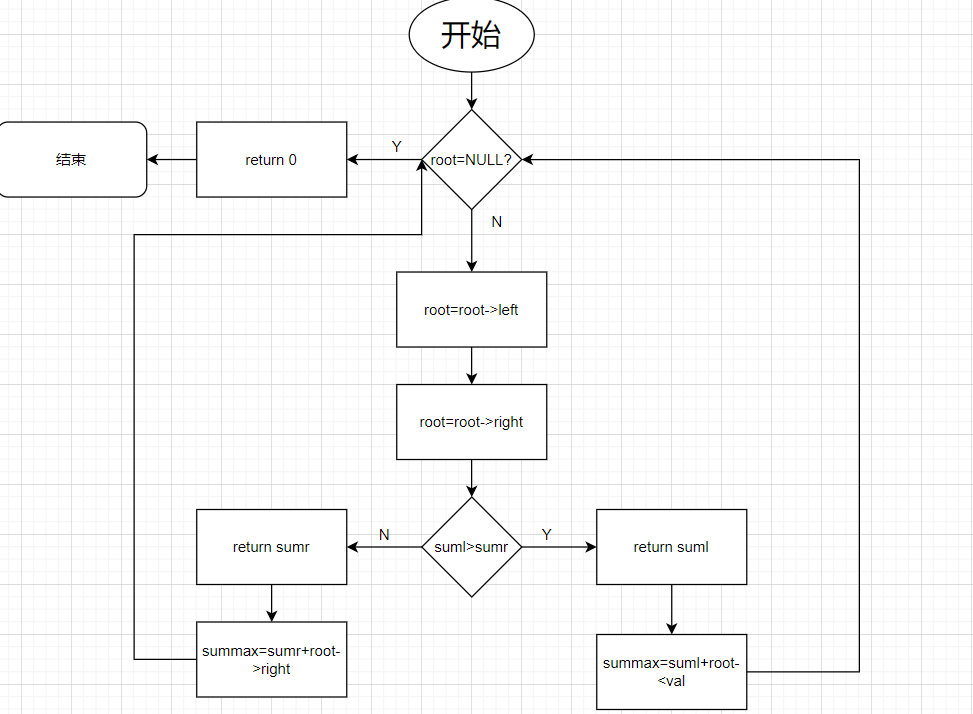
* 递归实现中序遍历



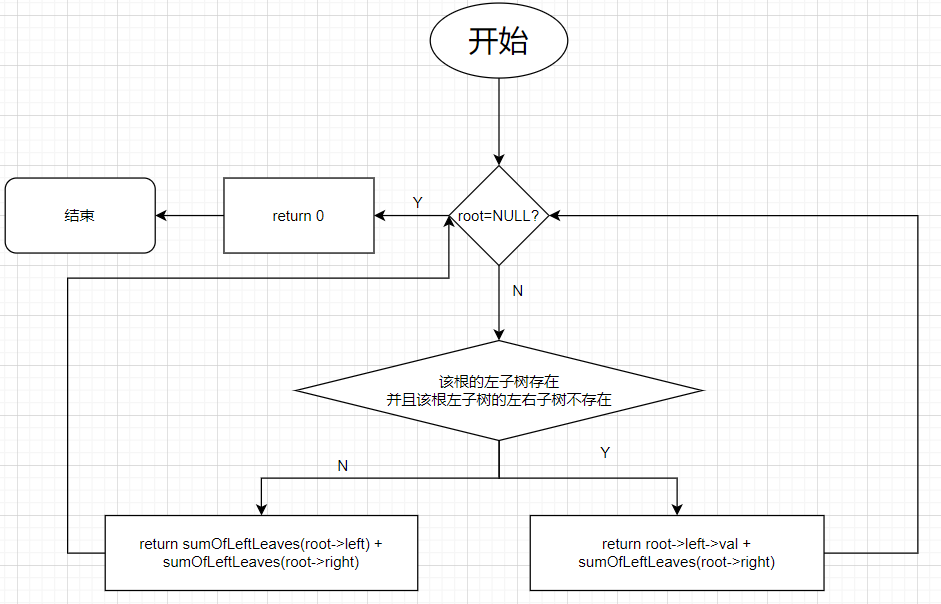
* 递归实现后序遍历



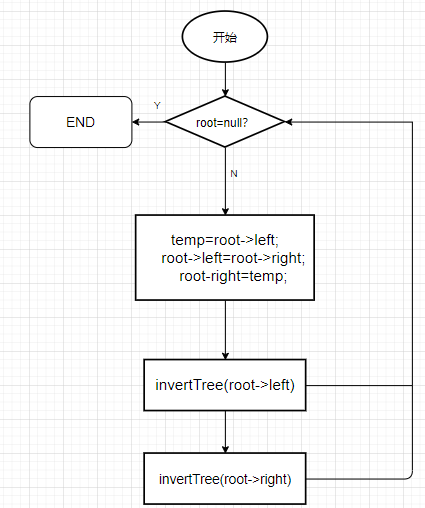
* 深度优先遍历来求取该二叉树的最大路径和



* 递归求取该二叉树的所有左子叶权重之和



* 通过递归求取该树的镜像，即翻转该二叉树

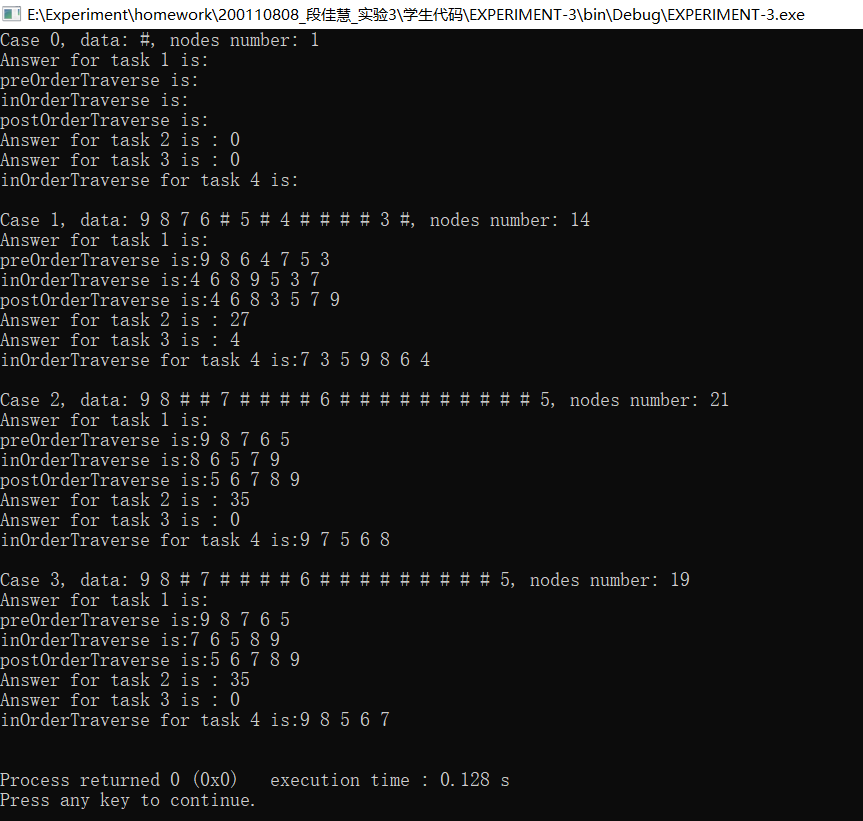


# 三、用户手册

如：(1)输入数据的方式；(2)实现各种功能的操作方式等。

1. 计算机直接从目标文件中读取数据
2. 用户直接运行，不需要手动实现各种操作

# 四、结果



# 五、总结

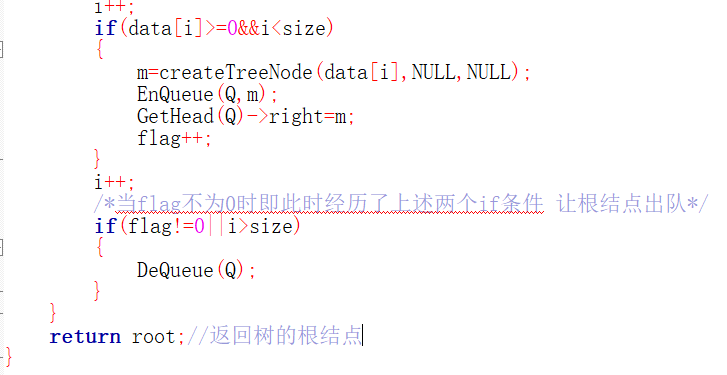
（该实验涉及到的数据结构和算法，以及遇到的问题和收获。）

本次实验主要涉及到 栈、队列、树、这几种数据结构

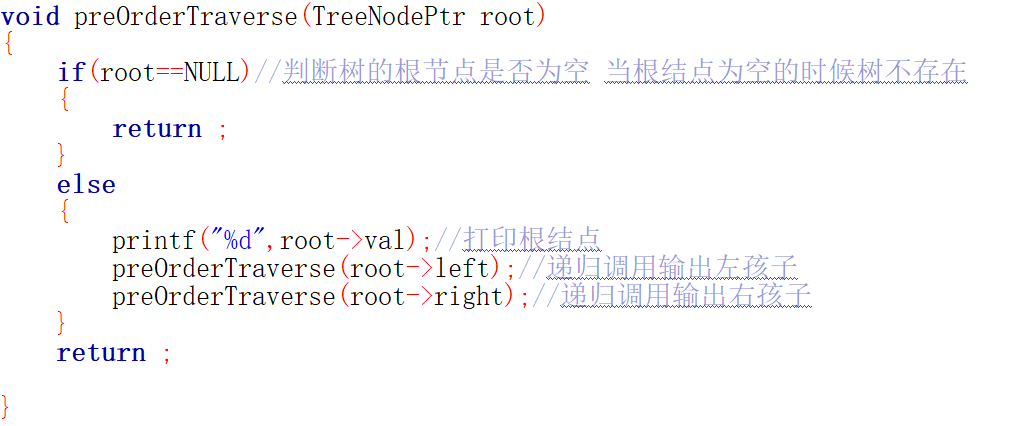
涉及到的算法：

1. 通过队列来实现层次遍历构建二叉树，并返回二叉树的头结点

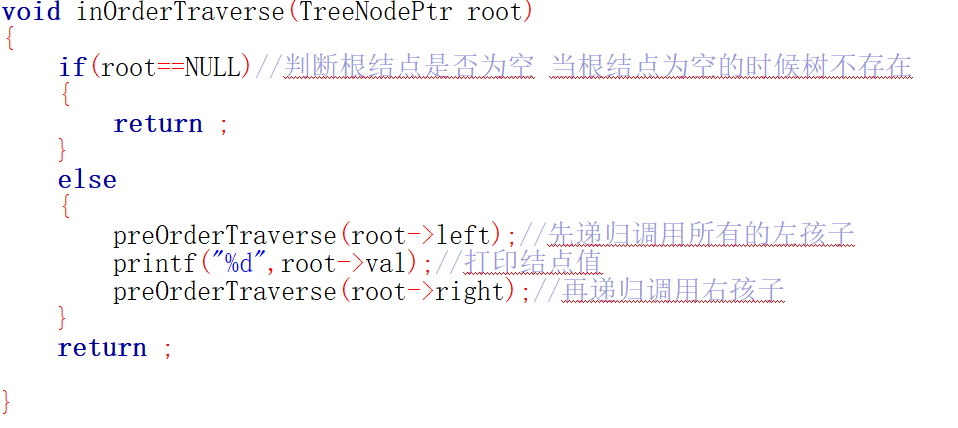




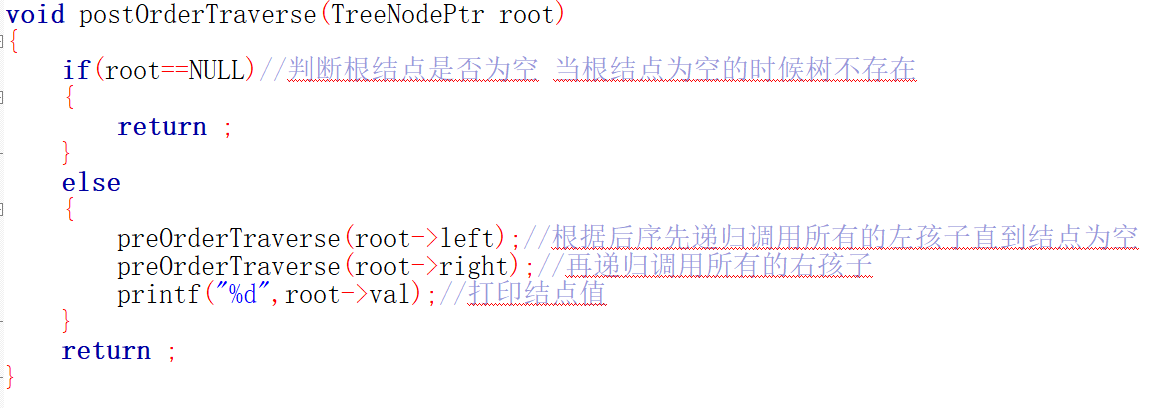
1. 前序遍历



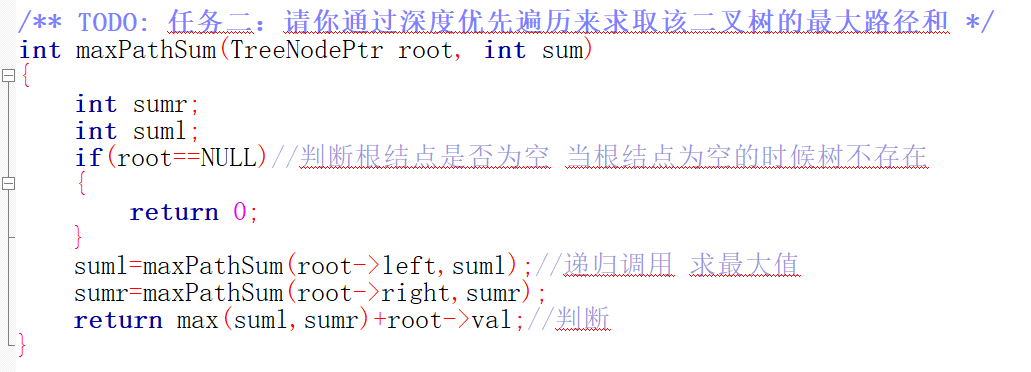
1. **中序遍历**



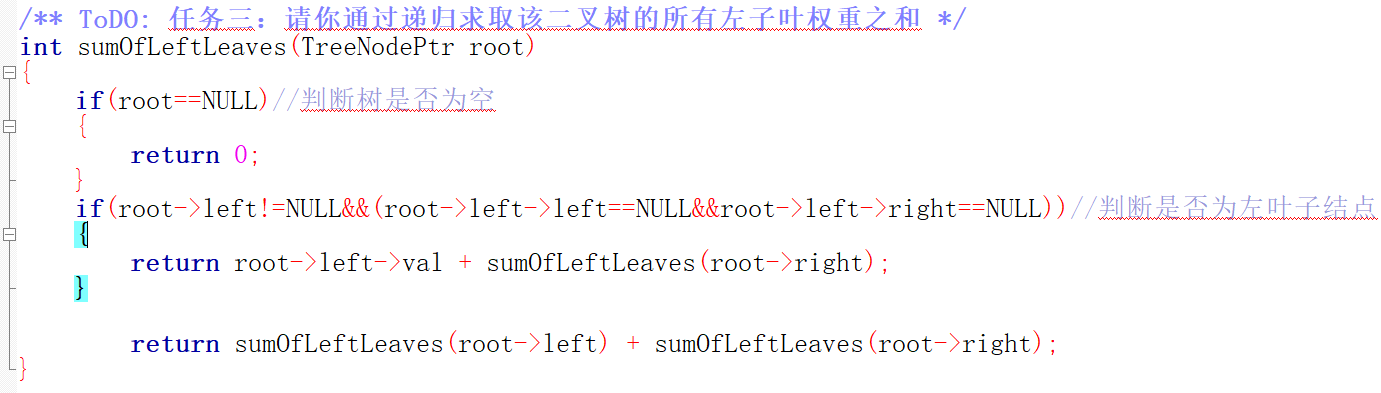
1. **后序遍历**



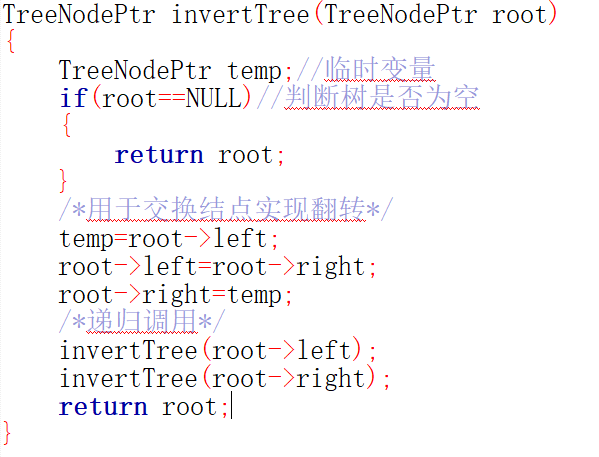
1. **通过深度优先遍历来求取该二叉树的最大路径和**



1. **通过递归求取该二叉树的所有左子叶权重之和**



1. **翻转二叉树，实现镜像**



在实验过程中遇到了几个问题，首先就是一开始对于树的理解不够导致刚上手写代码很多地方模棱两可，需要不停地翻看PPT和书本。其次在完成任务一的时候，无从下手，没有理解函数参数\*data的意思以及树和数组两个存储的关系，对于用队列来实现树也不够熟悉，导致在这一部分浪费了很长时间。另外在完成任务二的时候，没有想到正确合适的算法，想了很久，代码实现的过程也比较麻烦，同时在写代码的过程中也遇到了很多问题。

总的来说，这次实验使我对树的操作更加熟练，也使我更深刻地认识了这种数据结构在生活中的作用，在以后的学习过程中，我要多多注意代码规范，追求简洁、细致的编程方式，并且要考虑边界与特殊情况，力求更进一步。