|  |
| --- |
| 哈尔滨工业大学(深圳) |
| **《数据结构与算法》实验报告** |
|  |
| 实验三  树形结构及其应用  学 院: 机电工程与自动化学院   |  |  | | --- | --- | | 姓 名: | 葛旭 | | 学 号: | 190320517 | | 专 业: | 自动化 | | 日 期: | 2021-4-26 | |

# 一、问题分析

1.按先序序列建立二叉树，并输出该二叉树的中序遍历和后序遍历。

2.给定一棵二叉树的先序和中序遍历序列，构造该二叉树，并输出该二叉树的后序遍历。当输入的先序和中序遍历序列不能够构造出该二叉树时，应输出：

This pre-order traversal and mid-order traversal cannot construct a binary tree!

1. 给定一棵二叉树，返回该二叉树的深度，以及将该二叉树补全为满二叉树所需节点数。
2. 给定一棵二叉树，判断该二叉树是否为镜像对称(结构相同和元素相等)。

分析：利用结构体和指针，结构体中包含左右孩子，来建立二叉树，并根据先序后序中序来依次输出遍历节点。深度即二叉树的层数，用满二叉树节点数减去现在的节点数即为应该补充的节点数。

# 二、详细设计

## 2.1 设计思想

利用递归来进行先序中序和后序的遍历。利用递归来根据特殊先序创建二叉树，也可以根据先序和中序创建二叉树。

## 2.2 存储结构及操作

(1) 存储结构

树节点的结构体，包含节点存储的数据和左右孩子节点。

1. 涉及的操作
2. *// 由特殊先序序列建立二叉树*
3. Btree \* createBTree(char \*parent)
4. *// 计算二叉树节点数*
5. int nodesOfTree(Btree \* T)
6. *// 由先序序列和中序序列构建二叉树*
7. Btree\* build (char \*Preorder, char \*Inorder, int len)
8. *// 判断二叉树是否镜像对称*
9. int isSymmetric(Btree \*t1, Btree \*t2)
10. *//判断两字符串含有的字符是否相同*
11. int isTheSame(int len, char Preorder[MAX], char Inorder[MAX])
12. *//判断可否根据用户输入的前序和中序创建树*
13. Btree\* check()
14. *// 计算二叉树深度*
15. int depthOfTree(Btree \* T)
16. *// 计算距离构造满二叉树还需多少节点*
17. int nodesToCompleteBTree(Btree \* T)
18. *// 二叉树前序遍历*
19. int PreOrderTravel(Btree \*T)
20. *// 二叉树中序遍历*
21. int InOrderTravel(Btree \*T)
22. *// 二叉树后序遍历*
23. int PostOrderTravel(Btree \*T)

## 2.3 程序整体流程

# 绘图1三、用户手册

(1)输入数据的方式

首先输入特殊先序序列来创建二叉树，然后输出二叉树的中序和后序，随后电脑会计算二叉树的深度，以及补全为完全二叉树需要的节点数。然后检查是否可以由先序和中序来创建二叉树。随后对新的二叉树重复上述操作。

（2）实现各种功能的操作方式等。

只需要输入特殊先序序列，以及先序和后序，计算机会继续接下来的操作。

# 四、结果

# 五、总结

学习到了如何利用递归来建立二叉树，以及对二叉树进行先序中序后序的遍历，对镜像和二叉树深度的计算等。