**实 验 报 告**

|  |  |  |
| --- | --- | --- |
| **课程名称**：数据结构 | **班级**：软工21101 | **实验成绩**： |
| **实验名称**：二叉树的建立与应用（2） | **学号**：2105006207 | **批阅教师签字：董傲霜** |
| **实验编号**：实验八 | **姓名**：方福涛 | **实验日期：**2022 年 10 月 4日 |
| **指导教师**：董傲霜 | **组号**：wuyanzu | **实验时间**： 时 分－ 时 分 |

1. **实验设计思想**

说明二叉树的链式存储结构的设计思想。

1.创建一棵空二叉树；

2.对一棵存在的二叉树进行销毁；

3.根据输入某种遍历次序输入二叉树中结点的值，依序建立二叉树；

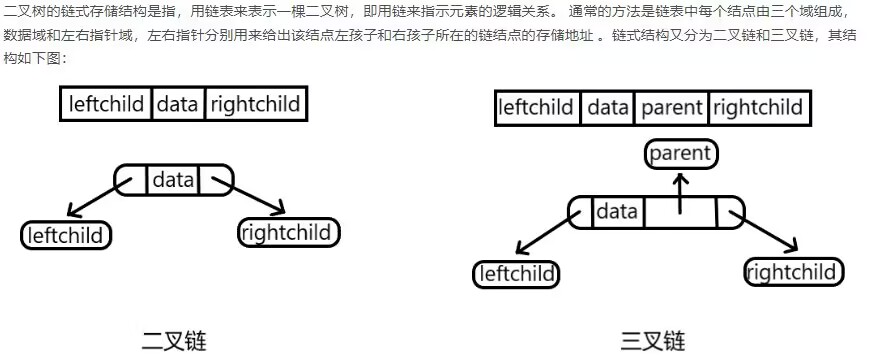
4.判断某棵二叉树是否为空；.求二叉树的深度；

5.求二叉树的根结点，若为空二叉树，则返回一特殊值；

6.二叉树的遍历，即按某种方式访问二叉树中的所有结点，并使每个结点恰好被访问

**二、程序说明**

1. 给出二叉树的链式存储结构定义及其属性含义的说明；



1. 给出你设计的函数的结构说明（函数，参数，执行的结果）；
2. 说明二叉树遍历操作实现步骤（处理步骤，可用文字也可画流程图）。

前序遍历——访问根结点的操作发生在遍历其左右子树之前。先根遍历

中序遍历——访问根结点的操作发生在遍历其左右子树之中（间）。中根遍历

后序遍历——访问根结点的操作发生在遍历其左右子树之后。后根遍历

**先序遍历：先根再左再右  
中序遍历：先左再根再右  
后序遍历，先左再右再根**

**三、实验环境**

Windows10、DEV-C++

**四、实验过程分析**

实验调试过程中的问题及改正的事件举例说明（只要调程序就不可能没有错误，现象、原因、修正方法，可以截图但不能过多）。

**易错点总结：**

**void createBiTree(BiTNode \*node)和void createBiTree(BiTNode \*&node)的区别**

void createBiTree(BiTNode \*node)是将结点的指针（地址）传递到函数中进行处理，而void createBiTree(BiTNode \*&node)是将结点指针的引用传递到函数处理。一般来说，如果node不为NULL的话，都是操作地址，个人感觉差别不是很大。但是在这里因为main函数初始化root是NULL，所以如果直接传递的是地址的话，会导致createBiTree执行完以后root任然为NULL（因为传过去的是地址的一个拷贝，如果root不为NULL的话，即使传过去的是地址的拷贝，因为我们操作的是地址所指向的值，所以关系不大，但是这里偏偏root为NULL，我们重新new了一个地址所以导致最后root任然是NULL）。所以将程序改写成传递指针的引用，问题很快就解决了。

**五、实验结果总结**

1、阐述二叉树链式存储结构上实现遍历操作的算法时间复杂度和空间复杂度。

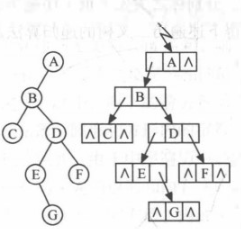
**时间复杂度：O(N)**

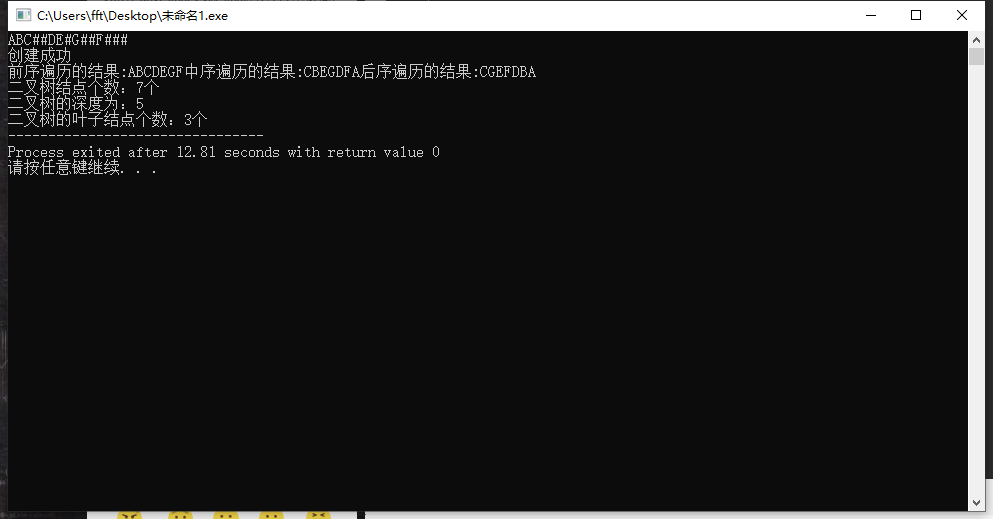
**空间复杂度：O(N)**

2、给出你的测试方法和测试截图（不能超过三个截图）；

案例：

输入：*ABC##DE#G##F###*





**六、附录**

1. 意见和建议（没有可不写）。

2. 思考题：

回答以下问题：

1. 二叉树的链式存储结构对比顺序存储结构有什么优缺点？

**顺序存储：**

优点：读取某个指定的节点的时候效率比较高O(0)

缺点：会浪费空间(在非完全二叉树的时候)

**链式存储：**

优点：读取某个指定节点的时候效率偏低O(nlogn)

缺点：相对二叉树比较大的时候浪费空间较少

1. n个结点的二叉树链式存储结构中，空指针的个数有多少个？？

空指针个数：n+1

**七、打分表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 项目 | 考核点 | 分数 | **得分** | 备注 |
| 程序 | 逻辑是否正确  程序可读性  创新点 | 50 | **0** |  |
| 报告完整性 | 实验过程阐述是否完整  测试数据设计是否合理  运行结果是否正确 | 40 | **0** |  |
| 调试问题及解决方法 | 是否对调试过程问题进行阐述 | 5 | **0** |  |
| 思考题目 | 回答是否正确 | 5 | **0** |  |
| 合计 |  | 100 | **0** |  |