数据结构与算法

实验报告

实验题目：\_\_\_\_\_\_\_\_\_\_\_\_\_\_二叉树\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

姓 名：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_李玮瀚\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

学 号：\_\_\_\_\_\_\_\_\_\_\_2020212265\_\_\_\_\_\_\_\_\_\_\_\_\_\_

日 期：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021.5.30\_\_\_\_\_\_\_\_\_\_\_\_\_

自我评分：\_\_\_\_\_\_\_\_\_【 A 】\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

自我评分说明：A+，A，B+，B，B-，C，D，分别对应分数95、90、85、80、75、70、60

诚信声明

本人郑重承诺：本实验程序和实验报告均是本人独立学习和工作所获得的成果。尽我所知，实验报告中除特别标注的地方外，不包含其他同学已经发表或撰写过的成果；实验程序中对代码工作的任何帮助者所作的贡献均做了明确的说明，并表达了谢意。

如有抄袭，本人原因承担因此而造成的任何后果。

特此声明。

签名：\_\_\_\_\_李玮瀚\_\_\_\_\_\_\_\_

日期：\_\_\_\_\_\_2021.5.30\_\_\_\_

程序引用说明

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 引用项 | 来源 | 相同代码行数 |
| 1 | 查找函数 | 课本 |  |
| 2 | 构造有序链表函数 | 互联网网址 |  |
| 3 | 打印 | 参考书 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| 小计 | | |  |

总代码行数\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_; 引用占比\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1、实验简介

根据二叉树的抽象数据类型的定义，使用二叉链表实现一个二叉树。

二叉树的基本功能：

1、二叉树的建立 2、前序遍历二叉树 3、中序遍历二叉树 4、后序遍历二叉树 5、按层序遍历二叉树 6、求二叉树的深度 7、求指定结点到根的路径 8、二叉树的销毁

2、程序框架

电脑萤幕的截图

描述已自动生成电脑萤幕的截图

描述已自动生成【实验程序共包含哪些函数，一一列举函数名和函数功能】

3、关键代码实现

文本

描述已自动生成3.1 Preorder\_Traversal(前序遍历)

利用栈,这种特殊的数据结构,可实现前序遍历,第一次进入循环将根节点放入栈中,循环中每次取栈顶一个节点,将其节点数据放入char类型的vector容器中,在判断该节点的左右子节点是否为NULL,如果不是NULL则将其子节点压入栈内,如此循环,直到栈内为空跳出循环,再将vector容器内遍历进行输出,即可得到前序遍历结果.

电脑萤幕的截图

描述已自动生成3.2 Postorder\_Traversal(后序遍历)

写后序遍历时,我进行了一些思考,是否能在前序遍历的基础上,进行一些改进来将其变为后续遍历,前序顺序为”根-左-右”,后序为”左-右-根”,其实就是将前序的左右反过来然后倒序输出即可,所以我写的后续算法基本根前序差不多,就是在判断子节点时顺序换了一下,然后再将temp1内出栈的元素,压入temp2中,最后将temp2中的元素一一出栈,将其数值放入vector容器中,打印输出即可

4、不足

代码的空间复杂度比较大,开辟了比较大的内存空间,依赖封装好的vector和stack容器,进行了一些操作

5、心得体会

了解并且掌握了二叉树的构建及其遍历等算法,没有使用递归进行遍历,减小了时间复杂度,提高了程序运行的效率.对二叉树这种算法的优缺点有了比较清晰的认识.