**理工学院计算机类课程实验报告**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 课程名称 | 数据结构 | | 实验名称 | 实验5 Project 二叉树 |
| 小组名称及成员 | 左康田、姜来、童明安、蔡杰铭、韦清秀 | | | |
| 专 业 | 软件工程 | | 年 级 | 2020 |
| 班 级 | 1班 | | 实验时间 | 2021.5.15-2021.5.17 |
| 教师评语（或成绩） | | 教师签字：  年 月 日 | | |
| **一、实验目的**  （1）了解C项目编写方式  （2）熟悉二叉树的基本运算以及算法与程序之间的关系 | | | | |
| **二、实验设备（工具、材料、硬软件）及要求**  1、设备：计算机一台、Dev C++  2、具体要求：  实现二叉树的基本算法：  （1）创建二叉树  （2）二叉树的遍历  （3）求二叉树的深度  （4）求二叉树的叶子数  **上述，各小组可根据实际情况，使用顺序存储或链式存储实现树的创建，在具体算法实现时可考虑递归和非递归两种写法。** | | | | |
| **三、实验过程**  要求：（1）列出实现的源代码、运行效果截图；（2）为源码添加必要的注释语句。另外，对实验进一步分析存在的问题提出解决办法，或总结实验所取得的经验均可写入报告中。**能力强者也可使用java语言实现代码。**  **/\*神仙打架\*/**  **#include <stdio.h>**  **#include <stdlib.h>**  **#include "btree.h"**  **int main(int argc, char \*argv[]) {**  **Bitree T;**  **T=createBitree(T);**  **Preordertraversal(T);**  **int cnt=0;**  **cnt=getdepth(T);**  **printf("\ndeep=%d",cnt);**  **int leaf=0;**  **leaf=getleaf(T);**  **printf("\nleaf=%d",leaf);**  **return 0;**  **}**  **typedef char Elemtype;**  **struct BNode{**  **Elemtype data;**  **struct BNode \*left;**  **struct BNode \*right;**  **};**  **typedef struct BNode BNode;**  **typedef struct BNode\* Bitree;**  **//先序的遍历**  **void Preordertraversal(Bitree T){**  **if(T!=NULL){**  **printf("%c",T->data);//先输出根结点**  **Preordertraversal(T->left);**  **Preordertraversal(T->right);**  **}**  **}**  **//中序的遍历**  **void Inordertraversal(Bitree T){**  **if(T!=NULL){**  **Inordertraversal(T->left);**  **printf("%c",T->data);//输出根结点**  **Inordertraversal(T->right);**  **}**  **}**  **//后序的遍历**  **void Postordertraversal(Bitree T){**  **if(T!=NULL){**  **Postordertraversal(T->left);**  **Postordertraversal(T->right);**  **printf("%c",T->data);//输出根结点**  **}**  **}**  **//创造二叉树**  **Bitree createBitree(Bitree T){**  **char x;**  **scanf("%c",&x);**  **if(x=='#'){**  **T=NULL;**  **}**  **else{**  **//造根**  **T=(Bitree)malloc(sizeof(BNode));**  **T->data=x;**  **//造左右**  **T->left=createBitree(T->left);**  **T->right=createBitree(T->right);**  **}**  **return T;**  **}**  **//二叉树的深度**  **int getdepth(Bitree T){**  **int leftnum=0,rightnum=0; //分别记录左右子树的深度**  **if(T==NULL){**  **return 0;**  **}**  **if(T->left==NULL&&T->right==NULL){**  **return 1;**  **}**  **else{**  **leftnum=getdepth(T->left);//求左子树的深度**  **rightnum=getdepth(T->right); //求右子树的深度**  **if(leftnum>=rightnum){**  **return leftnum+1;**  **}**  **else{**  **return rightnum+1;**  **}**  **}**  **}**  **//二叉树的叶子数**  **int getleaf(Bitree BT){**  **int l\_leaf=0,r\_leaf=0;//分别记录左右子树的叶子数**  **if(BT==NULL){**  **return 0;**  **}**  **if(BT->left==NULL&&BT->right==NULL){**  **return 1;**  **}**  **else{**  **l\_leaf=getleaf(BT->left);**  **r\_leaf=getleaf(BT->right);**  **return l\_leaf+r\_leaf;**  **}**    **}**        **本次作业小结：**  **完成进度较快，小组内配合度较高、分工明确、以合作的形式共同完成。**  **不太掌握二叉树的叶子数、深度的算法，在写代码的过程中这部分耗时较长；但对于二叉树的遍历思想的掌握程度还是比较高的。** | | | | |