

《数据结构》实验报告 6

学号: _____ 姓名: _____ 班级: _____ 成绩: _____

实验名称: 二叉树遍历的非递归实现

实验地点: 数学系机房

一、 评分标准:

1、 评分成绩为 A,B,C,D,E 五档, 满分为 A

1) 填空题: 每错 2 小题, 成绩降一档。

2) 简答题: 每小题回答不完整和错误处超过 2 条 (含 2 条), 总分降一档)

3) 程序填空: 每错两处, 总分降一档, 依次进行。

二、 填空题:

1、 已知一棵树的广义表表示为: A(B(E, F), C(G), D(H, I, J)), 则它对应的二叉树的先序遍历顺序为 (_____)。

2、 已知一棵二叉树的先序遍历是 ABCDEFGHIJ, 中序遍历是 BCDAFEHJIG, 则它的后序遍历是 (_____)。

3、 已知一棵二叉树的叶子结点个数为 15, 则其有 2 个孩子的结点个数为 (_____)

4、 如果对一棵有 n 个结点的完全二叉树按层次进行编号, 根结点标号为 1, 则第 i 个结点的左孩子的编号是 (_____)

三、 简答题

1、 已知一棵二叉树的后序遍历正好是下列函数调用 f(5,3) 的输出结果 (printf 对应于访问根结点), 请画出这棵二叉树, 并分别写出该二叉树先序、中序和后序遍历的结果。

```
f(int x, int y) {  
    if (x>y && y>0) f(x-y,y);  
    if (x<y && x>0) f(x,y-x);  
    printf("(%d, %d )\n", x,y);  
}
```

先序: _____

中序: _____

后序：

2、一棵二叉树的先序遍历和后序遍历一样,当且仅当这棵树满足什么特征?

答：

3、除了只有一个结点的二叉树外,有没有一棵二叉树的先序和中序遍历次序一样?有的话,请问这时树满足什么特征?没有的话,请给出原因?

答：

4、有 n 个结点的不同形状的二叉树有多少棵? (本题要求记住结论)

答：

5、已知一棵二叉树的先序遍历顺序为 ABCDE,请问这样的二叉树共有多少棵?如果二叉树有 n 个结点,给定先序遍历的顺序,则可画出多少棵不同的二叉树?

答：

6、把第 9 题中的“先序”依次改为“中序”、“后序”,结果如何?

答：

四、 实验填空：

本实验主要采用“二叉链表”来实现二叉树的存储表示,采用先(根)序遍历实现

(1) 建立二叉树的存储结构——二叉链表

(2) 求二叉树的树深。

由于“二叉链表”结构不是 C 语言已经提供的,因此需要首先定义“二叉链表”的结构体,实现先(根)序遍历算法,才能加以利用解决以上两个问题。

实验步骤如下：

- 1、定义“二叉链表”的结构
- 2、实现“二叉树”的一些操作
 - (1) 先（根）序遍历二叉树 Preorder
 - (2) 求二叉树的树深 BiTreeDepth
 - (3) 建立二叉树的存储结构 CreateBiTree
- 3、实现主程序 main

主程序主要是用来控制程序的执行过程，包括变量的初始化，调用以上三个操作，实现输出等。

4、程序的编译、链接

5、程序的测试

为了验证算法的有效性、体会二叉树操作的特点，需要采用一个实际的二叉树作用用例进行测试。测试前，应先写好该二叉树输入的顺序。用例二叉树如图 1 所示。

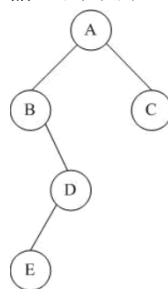


图 1

输入该二叉链表时，应按如下顺序：AB#DE###C##

操作步骤

```
#include <_____>

#include "_____"

#define TElemType char

#define NULL 0

typedef struct BiTNode{

    TElemType data;

    struct BiTNode *lchild,*rchild;

}BiTNode,*BiTree;

void Preorder(BiTree T)

{
```

```

        if(T!=NULL){

printf("%10c",T->data);

Preorder(T->_____);

Preorder(T->_____);

        }

}

int BiTreeDepth(BiTree T){

    int depth=0,hl,hr;

    if(T==NULL) depth=0;

    else{

        hl=BiTreeDepth(T->lchild);

        hr=BiTreeDepth(T->rchild);

        depth=1+(_____);

    }

    return depth;

}

BiTree CreateBiTree(BiTree T)

{

    char ch;

    printf("\n");

    printf("Input the data of node:");

    ch=getch();

    if(ch=='#'){

        T=NULL;

        printf("\nYou input is:#");

    }

    else{

```

```

        printf("\nYou input is:%c",ch);

        T=(BiTree)malloc(sizeof(BiTNode));

        T->data=_____;;

        T->lchild=NULL;

        T->rchild=_____;;

        T->lchild=CreateBiTree(_____);

        T->rchild=CreateBiTree(_____);

    }

    return T;

}

main()

{

    BiTree T;

    int dep;

    T=NULL;

    T=CreateBiTree(T);

    Preorder(T);

    dep=BiTreeDepth(T);

    printf("\nThe depth is:%d",_____);

}

```

评语:

任课教师: 赵宏庆 成绩: _____ 年 ____月____日