

实验五 树

一、实验目的

- 1、熟悉二叉树的基本操作。
- 2、熟悉递归在二叉树中的应用

二、定义数据结构

二叉链表

```
typedef struct BiTNode{  
TElemType data;  
struct BiTNode *lChild,*rChild;  
}BiTNode,*BiTree;
```

三、实验内容

- 1、实现下列树的基本操作

```
void createBiTree(BiTree &T);           //创建二叉树  
void preOrder(BiTree T);                //先序遍历  
void inOrder(BiTree T);                 //中序遍历  
void postOrder(BiTree T);               //后序遍历  
void levelTraversal(BiTree T);          //层序遍历  
int depth(BiTree T);                    //求树的深度  
int countNodes(BiTree T)                //后序遍历求节点数
```

测试数据（先序序列） 6 4 2 3 ##### 5 1 ## 7 ##（#代表空结点）

2、采用**非递归**的方法求出树的先序、中序、后序遍历。

```
void preOrder(BiTree T);    //先序遍历
```

```
void inOrder(BiTree T);    //中序遍历
```

```
void postOrder(BiTree T);    //后序遍历
```

3、利用递归的方式求解二叉树第 **K** 层节点的个数

```
int GetNumBitree(BiTNode* T,int k)
```

4、（附加题）判断二叉树是否为完全二叉树。

若设二叉树的深度为 **h**，除第 **h** 层外，其它各层 (**1~h-1**) 的结点数都达到最大个数，第 **h** 层所有的结点都连续集中在最左边，这就是完全二叉树。

测试数据：(先序序列, #代表空结点)

a) 非完全二叉树： 6 4 2 3 ##### 5 1 ## 7 ##

b) 完全二叉树： 6 4 2 3 ##### 6 ## 5 1 ## 7 ##