

实验六 树的延伸

一、实验目的

- 1、熟悉二叉线索树的基本操作。
- 2、深入了解递归在二叉树中的应用

二、定义数据结构

线索二叉树

```
typedef struct BiThrNode{  
    TElemType data;  
    struct BiThrNode *lChild,*rChild;  
    unsigned char ltag,rtag;  
}BiThrNode,*BiThrTree;
```

三、实验内容

- 1、建立中序**线索**二叉树，并实现对二叉树的中序遍历，并将结果输出。

```
void InThreading(BiThrTree T)
```

```
BiThrTree GetNext(BiThrTree P)
```

```
void InOrder(BiThrTree T)
```

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

- 2、仍沿用上面的数据，判断一个节点是否在一颗子树中（运用**二叉树**的定义即可）

该节点可以与该子树的根节点相等，也可以在左子树或者右子树中。

bool IsInTree(BiTree T, BiTree R)

3、（附加题）求二叉树中两个节点的最低公共祖先节点

例如：

1 和 7 的最低公共祖先是 5；

1 和 2 的最低公共祖先是 6

提示：

可利用上一题中的函数，对两个节点是否在一个子树中进行判断。可构建如下函数：

BiTree GetLastParent(BiTree pRoot, BiTree P1, BiTree P2)

其中 pRoot 是根节点，P1、P2 为比较节点