

问题描述

已知二叉树用二叉链表表示，其结点数据位字符型

- 1、从已知序列建立二叉树，空位用 0 表示，建立二叉链表；
- 2、用逆中序遍历输出该二叉树(每行一个结点，同层列对齐，最左为树根)；
- 3、分别输出中序遍历、后序遍历、按层次遍历；
- 4、求二叉树高度；
- 5、计算二叉树叶子结点个数。

算法思路

对于树的常见操作：

建立二叉链表来存储表示一棵树；

中序、后序、逆中序遍历；

采用队列的数据结构按层次遍历；

比较最长路径计算树的高度；

数叶结点（无左右子树的结点）个数计算叶结点个数。

算法描述

1. 表示二叉树的数据结构

常用二叉链表表示并储存一棵树。

链表中每个结点由三个域组成：除了数据域外，还有两个指针域，分别指向该结点左右孩子结点。结点的存储结构为：

lchild	data	rchild
--------	------	--------

对于本程序，二叉链表建立为：

```
typedef struct Binarytree
{
    char a;
    struct Binarytree *lc;
    struct Binarytree *rc;
}BTN;
```

2. 递归遍历

常见的递归遍历有六种，本程序要求中序和后序遍历。具体代码详见附件

3. 按层次遍历

按层次遍历涉及每层结点的存储与输出问题。为保存同一层各结点并输出，考虑采用 FIFO 的队列作为临时存储。

4. 逆中序法输出树形

树形等同于逆中序遍历逆时针旋转九十度放置。输出时需要考虑输出空格个数和换行条件，即每行一个元素，前方空格数等同于层数整数倍。

源程序及驱动程序

见附件

测试数据

输入序列为：A B D 0 G 0 0 0 C E 0 0 F 0 0

程序运行结果为：

```
原始序列参考教材为：
A B D 0 G 0 0 0 C E 0 0 F 0 0
逆中序遍历树形为：
      F
     /
    C
   / \
  A   E
   \
   B
    \
    D
     \
     G

中序遍历序列为：
D G B A E C F
后序遍历序列为：
D G B E C F A
按层次遍历序列为：
A B C D E F G
高度为：
4
结点个数为：
3

请按任意键继续. . .
```

结果分析和结论

通过逆中序遍历输出可以很方便地看到树形图，利于学习分析；

各种递归遍历方式异同可类比；

输出结果正常；

递归算法更适合这种可以分为一步一步的遍历问题。迭代在这里显得并不是十分简洁；

心得体会

值得思考的点有几个：实现递归遍历、如何让树形图更加直观？（如添加到表格，连线等）、二叉链表输入简化等等。