# 线性结构

## 2.3队列

先进先出的数据结构，只能在一端进行插入，另外一端进行删除

数据实现请参考chap02文件夹

# 树

## 3.1

#### 3.1.1树的相关术语

树基本概念：n≥0个节点构成的集合；

根：类似于树根，对于一颗非空树，只有一个

子树：除根节点外，其余节点可分为m个互不相交的有限集，其中每一颗树又是可以作为一棵树

**判断一个集合是否为树，看他的子树是否相连并且每个节点的父节点只有一个**

**对于一个N节点构成的树，有N-1条边**

树的基本术语：

节点的度：节点的子树个数；

树的度：取树中最大的节点的度最为整棵树的度

叶子：度为0的节点

父节点：有子树的节点是其子树的根节点的父节点

子节点（孩子节点）：b是a的父子点，那么a是b的子节点

兄弟节点：拥有同一个父节点的节点集合互为兄弟节点

路径与路径长度：从n1—nk构成的节点序列（其中ni是ni+1）的父节点）称为路径

路径长度则是边的个数；

祖先节点：从树根到某一结点n所经过的结点都是n的祖先结点

子孙节点：某一结点的子树所有结点

结点的层次：规定根节点是一层，其余结点的层是其父节点的层数+1

树的深度：取结点层次的最大值

#### 3.1.2树的数据结构表示

儿子-兄弟表示法：申请一个含有两个指针的结构体，一个指向第一个儿子，一个指向它的兄弟

**这里注意到我们构造出来的也是一棵树，这棵树的特点是结点的度最大为2，这种树我们称为二叉树**

#### 3.1.3二叉树

斜二叉树：只有左半边或者右半边的二叉树；

满二叉树：除了叶子结点，每个结点的度都2；

完全二叉树：对二叉树进行从上到下，从左到右进行编号，每一个都对应上满二叉树的结点的编号

**二叉树性质：**

1.第n层的最大节点数：2n-1

2.深度为k的二叉树最大结点总数：2k-1

3.对于非空的二叉树T，若n0表示叶子个数，n2表示度为2的结点数，，那么n0=n2+1

（证明：若n1表示度为1的结点树，那么n0+n1+n2-1是变得总数，也等于0\*n0+1\*n1+2\*n2）

**遍历方法：**

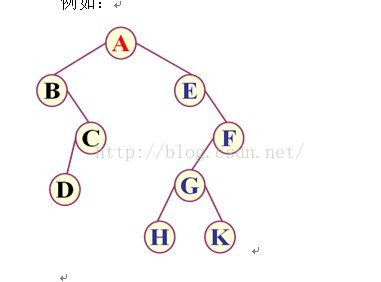
先序遍历：根-左-右

中序遍历：左-根-右

后序遍历：左-右-根

层次遍历：从上到下

示例：



先序遍历：ABCDEFGHK

中序遍历：BDCAEHGKF

后序遍历：DCBHKGFEA

(我们以中序遍历为例：先访问B，此时B没有左子树故停止，假如b还有左子树，则应继续，按照这个原则，输出B；来到c，c有左子树d，而d没有左子树，故先输出d，再输出c，其余以此类推)