第五章 树和二叉树

# 1 树的概念，要求达到“识记”层次

## 树的定义和表示方法

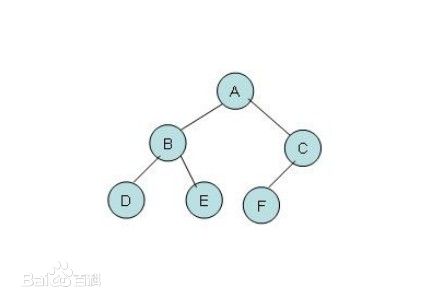
### 树的定义

树（Tree）是n（n≥0）个结点的有限集T。它或是空集（空树即n=0），或者是非空集。对于任意一颗非空树：

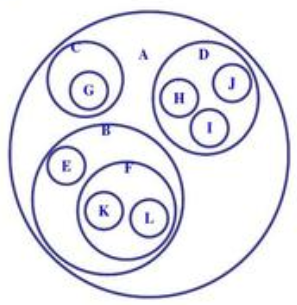
1. 有且仅有一个特定的成为根（Root）的结点；
2. 当n>1时，其余的结点可分为m（m>0）个互不相交的有限集T1，T2，…，Tm，其中每个集合本身又是一颗树，并称为根的子树。

### 树的表示方法

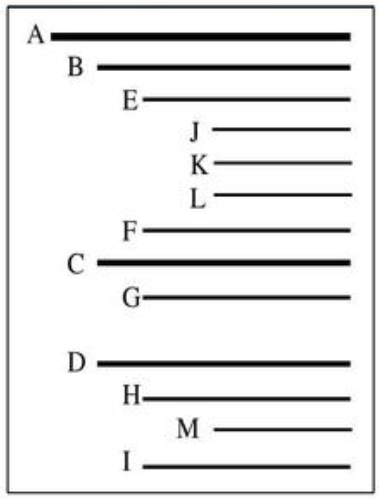
1. 树形图



1. 嵌套集合



1. 凹形表



1. 广义表

（A（B（E，（F（J，k））），C（G），D（H，I）））

## 树的常用术语及其含义

1. 树的结点包含一个数据元素及若干个指向其子树的分支。一个结点拥有的子树数称为该结点的度。
2. 一棵树中结点的最大度数称为该树的度（Degree）。
3. 度数为零的结点称为叶子（Leaf）结点或者终端结点。
4. 度数不为零的结点称为非终端结点或分支结点。
5. 除根结点之外，分支结点也称为内部结点，而根节点又称为开始结点。
6. 树中某个结点子树的根称为该结点的孩子（Child），相应的，该结点称为孩子结点的双亲（Parent）或父结点。
7. 若在一棵树中存在着一个结点序列k1，k2，…，kj使得ki是ki+1的父节点（1≤i≤j），则称该结点序列是从k1到kj的一条路径。
8. 若树中结点ki到kj存在一条路径，则称结点ki是kj的祖先，结点kj是ki的子孙。
9. 树中结点的层次（Level）是从根开始算起的，根为第一层，其余结点的层次等于其双亲结点的层树加1。
10. 树中结点的最大层次称为树的深度（Depth）或高度。
11. 如果将树中结点的各子树看成是从左至右依次有序且不能交换，则称该树为有序树，否则称为无须树。
12. 森林（Forest）是m（m≥0）棵互不相交的树的集合。

# 2 二叉树的概念，要求达到“领会”层次

## 2.1 二叉树的递归定义

二叉树（Binary Tree）是n（n≥0）个结点的有限集合，它的每个结点之多只有两棵子树。它或者是空集，或者是由一个根结点以及两颗互不相交的分别称作这个根的左子树和右子树的二叉树组成。

## 2.2 二叉树的性质及其证明，两种特殊形式的二叉树

1. 在二叉树的第i层上之多有2i-1个结点（i ≥ 1）。

利用书写归纳法证明如下：

1. 当 i=1 时，只有一个根结点，即2i-1=20=1.命题成立。
2. 假设对所有的j（1≤j＜i），命题成立，即第j层上最多有2j-1个结点。
3. 由归纳假设，第i-1层上至多有2i-2个结点。由于二叉树的每个结点的度数至多为2，因此，在第i层上的结点数至多是第i-1层上最大结点数的2倍，即2 X 2i-2= 2i-1 个结点，所以命题成立。

2. 深度

## 2.3 二叉树的顺序存储和链式存储结构

# 3 二叉树的运算，要求达到“综合应用”层次

## 3.1 二叉链表生产

## 3.2 二叉树的递归遍历算法和非递归遍历算法

## 3.3 二叉树的应用

# 4 线索二叉树，要求达到“简单应用”层次

## 4.1 二叉树线索化的含义，线索二叉树结点的表示方法

## 4.2 对给定二叉树进行线索化的思想和实现

## 4.3 二叉线索链表上的运算：查照某结点的后继节点和线索二叉树的遍历

# 5 树和森林，要求达到“领会”层次

## 5.1 树的三种存储结构表示方法

## 5.2 树，森林和二叉树之间的相互转换

## 5.3 树和森林的遍历

# 6 哈夫曼树及其应用，要求达到“简单应用”层次

## 6.1 最优二叉树的概念

## 6.2 哈夫曼算法的实现

## 6.3 编码，前缀编码，哈夫曼编码的概念；根据最优二叉树构造对应的哈夫曼编码