



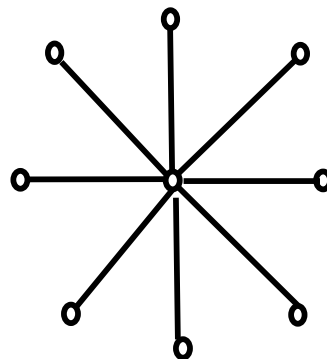
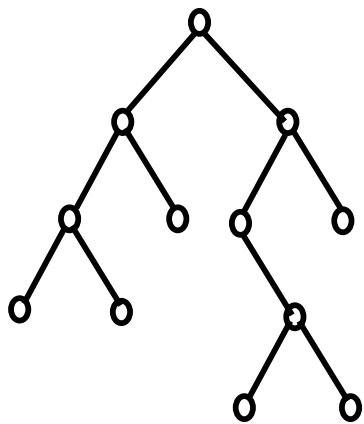
## 主要内容

- 无向树及其性质
- 生成树
- 根树及其应用



**定义12.1** 连通无回路的无向图称为**无向树**, 简称**树**. 每个连通分支都是树的无向图称为**森林**. 平凡图称为**平凡树**. 在无向树中, 悬挂顶点称为**树叶**, 度数大于或等于2的顶点称为**分支点**.

例



星形树



**定理12.1** 设 $G=\langle V, E \rangle$ 是 $n$ 阶 $m$ 条边的无向图，则下面各命题是等价的：

- (1)  $G$  是树
- (2)  $G$  中任意两个顶点之间存在惟一的路径.
- (3)  $G$  中无回路且  $m=n-1$ .
- (4)  $G$  是连通的且  $m=n-1$ .
- (5)  $G$  是连通的且  $G$  中任何边均为桥.
- (6)  $G$  中没有回路，但在任何两个不同的顶点之间加一条新边后所得图中有惟一的一个含新边的圈.



证 (1) $\Rightarrow$ (2). 若路径不惟一, 必有回路.

(2) $\Rightarrow$ (3). 若 $G$ 中有回路, 则回路上任意两点之间的路径不惟一. 对 $n$ 用归纳法证明 $m=n-1$ .

当 $n=1$ 时成立. 设 $n \leq k$ 时成立, 证 $n=k+1$ 时也成立: 任取一条边 $e$ ,  $G-e$ 有且仅有两个连通分支 $G_1, G_2$ .  $n_i \leq k$ , 由归纳假设得 $m_i = n_i - 1, i=1, 2$ . 于是,

$$m = m_1 + m_2 + 1 = n_1 + n_2 - 2 + 1 = n - 1.$$

(3) $\Rightarrow$ (4). 只需证明 $G$ 连通. 用反证法. 否则 $G$ 有 $s (s \geq 2)$ 个连通分支, 它们都是树. 于是, 有 $m_i = n_i - 1$ ,

$$m = \sum_{i=1}^s m_i = \sum_{i=1}^s n_i - s = n - s \quad (s \geq 2)$$

这与 $m=n-1$ 矛盾.



(4) $\Rightarrow$ (5). 只需证明 $G$ 中每条边都是桥. 下述命题显然成立:  $G$ 是 $n$ 阶 $m$ 条边的无向连通图, 则 $m \geq n-1$ .

$\forall e \in E, G-e$ 只有 $n-2$ 条边, 由命题可知 $G-e$ 不连通, 故 $e$ 为桥.

(5) $\Rightarrow$ (6). 由(5)易知 $G$ 为树. 由(1) $\Rightarrow$ (2)知,  $\forall u, v \in V (u \neq v)$ ,  $u$ 到 $v$ 有唯一路径, 加新边 $(u, v)$ 得惟一的一个圈.

(6) $\Rightarrow$ (1). 只需证明 $G$ 连通, 这是显然的.



**定理12.2** 设 $T$ 是 $n$ 阶非平凡的无向树，则 $T$ 中至少有两片树叶.

证 设 $T$ 有 $x$ 片树叶，由握手定理及定理10.1可知，

$$2(n-1) = \sum d(v_i) \geq x + 2(n-x)$$

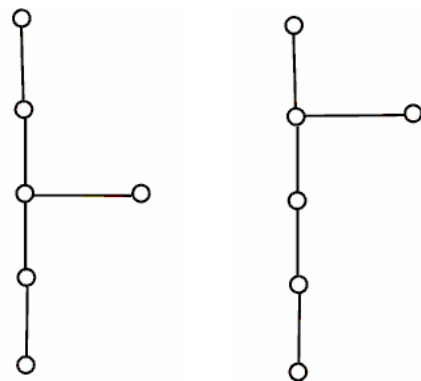
解得  $x \geq 2$ .

**例1** 已知无向树 $T$ 中有1个3度顶点，2个2度顶点，其余顶点全是树叶，试求树叶数，并画出满足要求的非同构的无向树.

解 设有 $x$ 片树叶， $n = 3+x$ .

$$2m = 2(n-1) = 2 \times (2+x) = 1 \times 3 + 2 \times 2 + x$$

解出 $x = 3$ ，故 $T$ 有3片树叶.



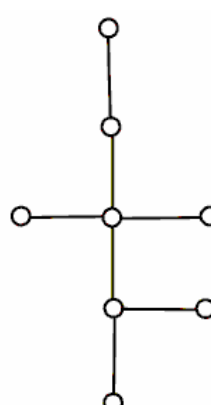
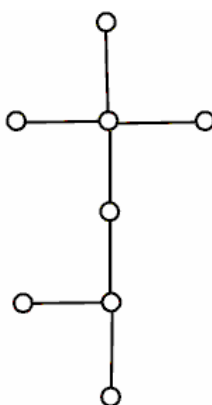
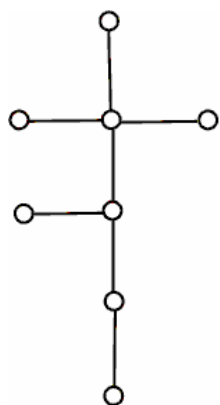


**例2** 已知无向树 $T$ 有5片树叶，2度与3度顶点各1个，其余顶点的度数均为4，求 $T$ 的阶数 $n$ ，并画出满足要求的所有非同构的无向树.

解 设 $T$ 的阶数为 $n$ ，则边数为 $n-1$ ，4度顶点的个数为 $n-7$ .

由握手定理，  $2m = 2(n-1) = 5 \times 1 + 2 \times 1 + 3 \times 1 + 4(n-7)$ ,

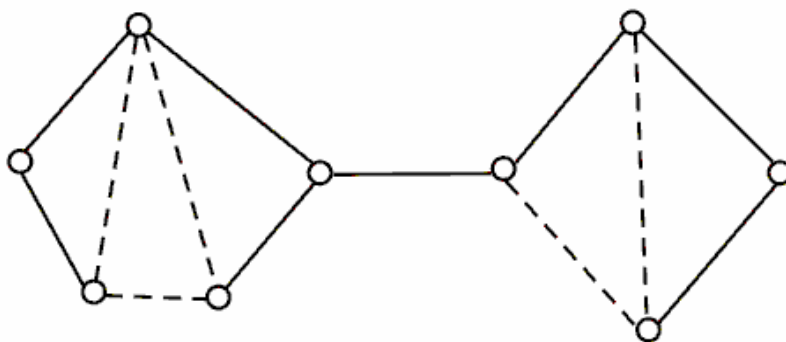
解出 $n = 8$ ，4度顶点为1个.





**定义12.2** 如果无向图 $G$ 的生成子图 $T$ 是树，则称 $T$ 是 $G$ 的**生成树**. 设 $T$ 是 $G$ 的生成树， $G$ 的在 $T$ 中的边称为 $T$ 的**树枝**，不在 $T$ 中的边为 $T$ 的**弦**. 称 $T$ 的所有弦的导出子图为 $T$ 的**余树**，记作 $\bar{T}$ .

**例**







**定理12.3** 无向图 $G$ 有生成树当且仅当 $G$ 连通.

证 必要性显然. 证充分性. 若 $G$ 中无回路, 则 $G$ 为自己的生成树. 若 $G$ 中含圈, 任取一圈, 随意地删除圈上的一条边; 若仍有圈, 再任取一个圈并删去这个圈上的一条边, 重复进行, 直到最后无圈为止. 最后得到的图无圈 (当然无回路)、连通且是 $G$ 的生成子图, 因而是 $G$ 的生成树.

这个产生生成树的方法称为**破圈法**.

**推论**  $G$ 为 $n$ 阶 $m$ 条边的无向连通图, 则 $m \geq n-1$ .



**定义12.3** 设无向连通带权图 $G=\langle V, E, W \rangle$ ,  $T$ 是 $G$ 的一棵生成树,  $T$ 的各边权之和称为 $T$ 的**权**, 记作 $W(T)$ .  $G$ 的所有生成树中权最小的生成树称为 $G$ 的**最小生成树**.

### 避圈法 (Kruskal)

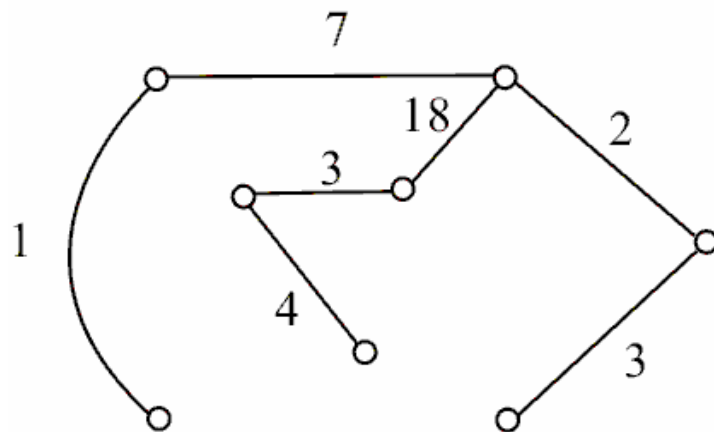
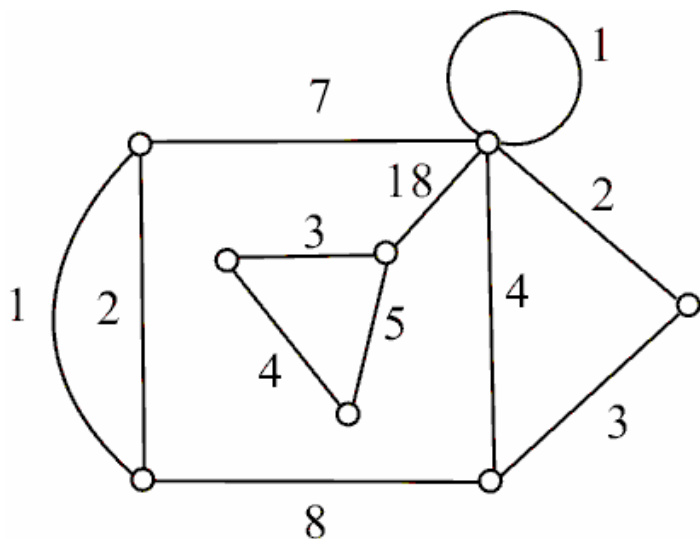
输入: 连通图 $G=\langle V, E, W \rangle$

输出:  $G$ 的最小生成树 $T$

1. 将 $G$ 中非环边按权从小到大排列:  $W(e_1) \leq W(e_2) \leq \dots \leq W(e_m)$ .
2. 令 $T \leftarrow \{e_1\}$ ,  $i \leftarrow 2$ .
3. 若 $e_i$ 与 $T$ 中的边不构成回路, 则令 $T \leftarrow T \cup \{e_i\}$ .
4. 若 $|T| < n-1$ , 则令 $i \leftarrow i+1$ , 转3.



**例4** 求图的一棵最小生成树.

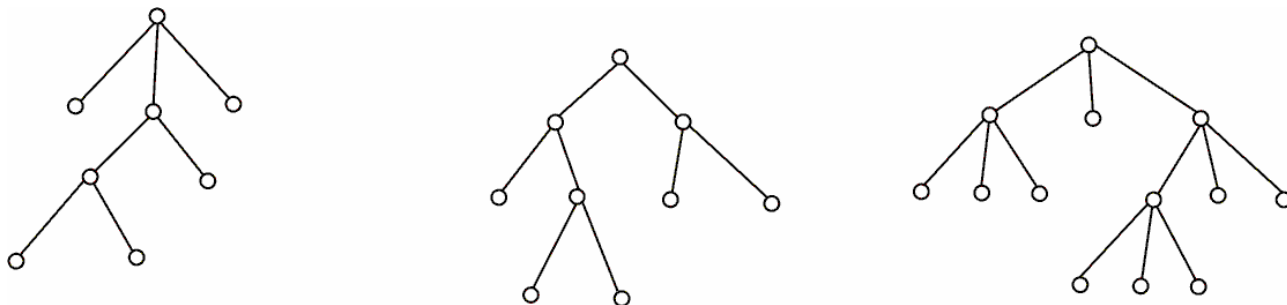


$$W(T)=38$$



**定义12.4** 若有向图的基图是无向树, 则称这个有向图为**有向树**. 一个顶点的入度为0、其余顶点的入度为1的有向树称为**根树**. 入度为0的顶点称为**树根**, 入度为1出度为0的顶点称为**树叶**, 入度为1出度不为0的顶点称为**内点**, 内点和树根统称为**分支点**. 从树根到顶点 $v$ 的路径的长度(即, 路径中的边数)称为 $v$ 的**层数**. 所有顶点的最大层数称为**树高**.

根树的画法——树根放上方, 省去所有有向边上的箭头  
**例**





**定义12.5** 设 $T$ 为一棵非平凡的根树,  $\forall v_i, v_j \in V(T)$ , 若 $v_i$ 可达 $v_j$ , 则称 $v_i$ 为 $v_j$ 的**祖先**,  $v_j$ 为 $v_i$ 的**后代**; 若 $v_i$ 邻接到 $v_j$ , 则称 $v_i$ 为 $v_j$ 的**父亲**,  $v_j$ 为 $v_i$ 的**儿子**. 若 $v_j, v_k$ 的父亲相同, 则称 $v_j$ 与 $v_k$ 是**兄弟**.

将根树 $T$ 中层数相同的顶点都标定次序, 称 $T$ 为**有序树**.

根树的分类:

- (1) 若 $T$ 的每个分支点至多有 $r$ 个儿子, 则称 $T$ 为 **$r$ 叉树**.
- (2) 若 $T$ 的每个分支点都恰好有 $r$ 个儿子, 则称 $T$ 为 **$r$ 叉正则树**.
- (3) 若 $T$ 是 **$r$ 叉正则树**, 且所有树叶的层数相同, 则称 $T$ 为 **$r$ 叉完全正则树**.

有序的 **$r$ 叉树**,  **$r$ 叉正则树**,  **$r$ 叉完全正则树**分别称作 **$r$ 叉有序树**,  **$r$ 叉正则有序树**,  **$r$ 叉完全正则有序树**.



**定义12.6** 设 $T$ 为一棵根树,  $\forall v \in V(T)$ , 称 $v$ 及其后代的导出子图 $T_v$ 为以 $v$ 为根的**根子树**.

2叉正则有序树的每个分支点的两个儿子导出的根子树分别称为该分支点的**左子树**和**右子树**.

**定义12.7** 设2叉树 $T$ 有 $t$ 片树叶 $v_1, v_2, \dots, v_t$ , 权分别为 $w_1, w_2, \dots, w_t$ , 称  $\sum_{i=1}^n w_i l_i$  为 $T$ 的权, 其中 $l_i$ 是 $v_i$ 的层数. 在所有有 $t$ 片树叶, 带权 $w_1, w_2, \dots, w_t$ 的2叉树中, 权最小的2叉树称为**最优2叉树**.



## Huffman算法

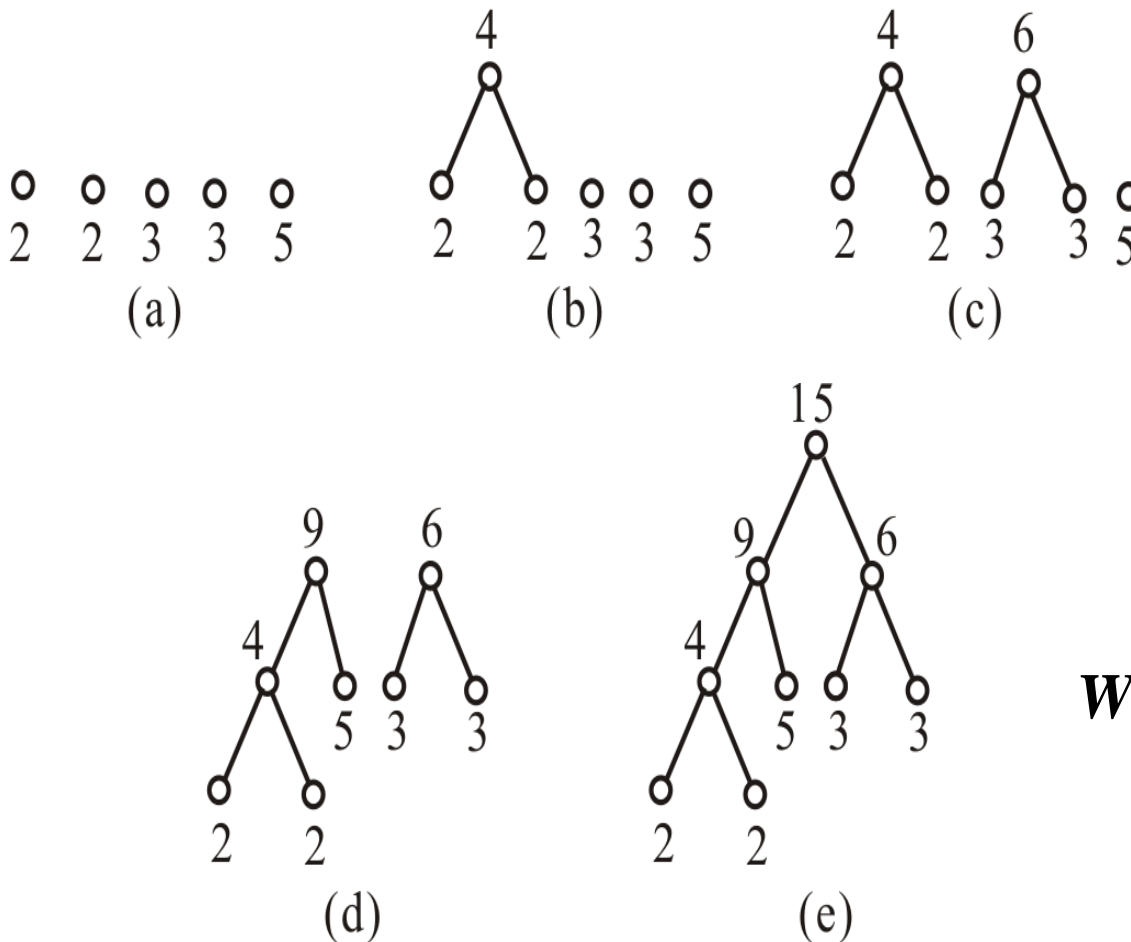
输入: 实数 $w_1, w_2, \dots, w_t$

输出: 最优二叉树

1. 作 $t$ 片树叶, 分别以 $w_1, w_2, \dots, w_t$ 为权.
  2. 在所有入度为0的顶点(不一定是树叶)中选出两个权最小的顶点, 添加一个新分支点, 它以这2个顶点为儿子, 其权等于这2个儿子的权之和.
  3. 重复2, 直到只有1个入度为0的顶点为止.
- $W(T)$ 等于所有分支点的权之和.



**例 5** 求权为2, 2, 3, 3, 5的最优树.  
解



$$W(T)=34$$





**定义12.8** 设 $\alpha_1\alpha_2\ldots\alpha_{n-1}\alpha_n$ 是长为 $n$ 的符号串, 称其子串 $\alpha_1, \alpha_1\alpha_2, \ldots, \alpha_1\alpha_2\ldots\alpha_n$ 为该符号串的**前缀**. 设 $A=\{\beta_1, \beta_2, \ldots, \beta_m\}$ 是一个符号串集合, 若 $A$ 的任意两个符号串都互不为前缀, 则称 $A$ 为**前缀码**. 由0-1符号串构成的前缀码称作**二元前缀码**.

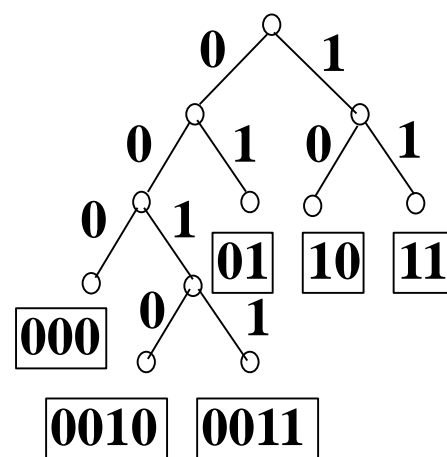
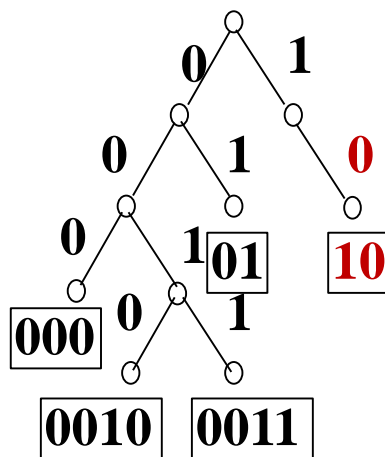
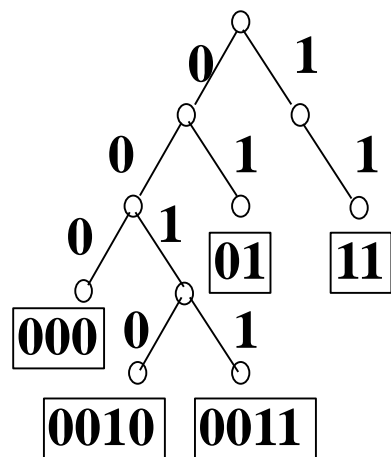
**例**  $\{1, 00, 011, 0101, 01001, 01000\}$ 为前缀码.

$\{1, 00, 011, 0101, 0100, 01001, 01000\}$ 不是前缀码.



用2叉树产生二元前缀码

例



一棵正则2叉树产生惟一的前缀码（按左枝标0，右枝标1）



设符号 $A_i$ 在传输中出现的频率为 $p_i$ , 二元前缀码的长为 $l_i$ ,  $1 \leq i \leq t$ , 传输 $m$ 个符号需要 $m \sum_{i=1}^t p_i l_i$  个二进制位.  $\sum_{i=1}^t p_i l_i$  最小的二元前缀码称作**最佳前缀码**. 最佳前缀码可用Huffman算法计算以频率为权的最优二叉树产生.

**例6** 设在通信中, 八进制数字出现的频率如下:

0: 25%	1: 20%	2: 15%	3: 10%
4: 10%	5: 10%	6: 5%	7: 5%

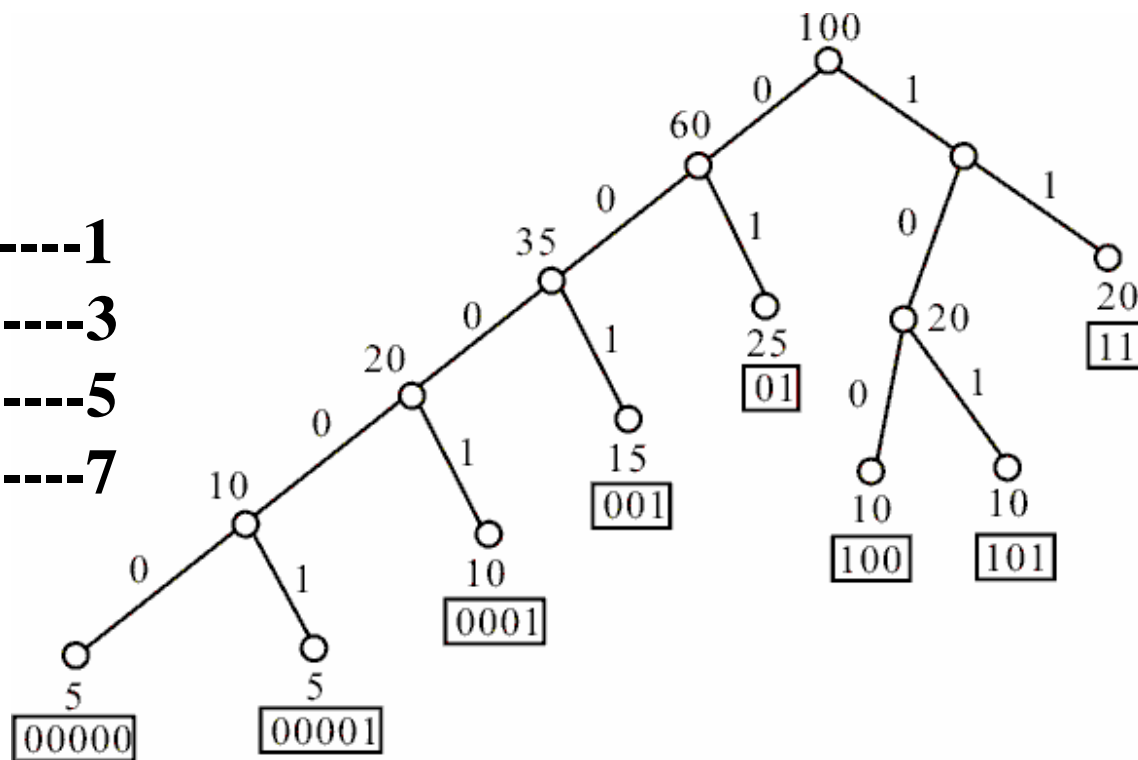
求传输它们的最佳前缀码, 并求传输 $10^n$  ( $n \geq 2$ ) 个按上述比例出现的八进制数字需要多少个二进制数字? 若用等长的(长为3)的码字传输需要多少个二进制数字?



解 传输100个八进制数字中各数字出现的个数,即以100乘各频率为: 25, 20, 15, 10, 10, 10, 5, 5, 以它们为权构造最优二叉树.

最佳前缀码

01-----0	11-----1
001-----2	100-----3
101-----4	0001-----5
00000-----6	00001-----7



$W(T)=285,$

传输 $10^n (n \geq 2)$ 个八进制数字,用最佳前缀码需 $2.85 \times 10^n$ 位,  
用等长码需 $3 \times 10^n$ 位.

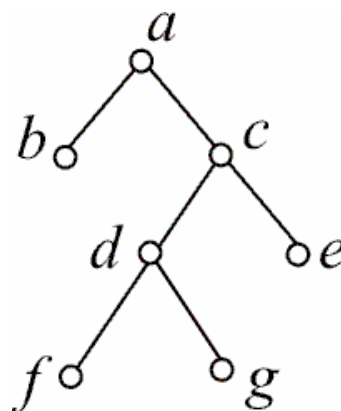


**行遍(周游)**有序树——对每个顶点访问且仅访问一次.  
对2叉有序正则树的周游方式:

- ① **中序行遍法**. 访问次序为: 左子树、根、右子树
- ② **前序行遍法**. 访问次序为: 根、左子树、右子树
- ③ **后序行遍法**. 访问次序为: 左子树、右子树、根

**例** 用中序, 前序, 后序行遍法访问的结果分别为:

$b \underline{a} (f \underline{d} g) \underline{c} e,$   
 $\underline{a} b (\underline{c} (d f g) e),$   
 $b ((f g \underline{d}) e \underline{c}) \underline{a}$



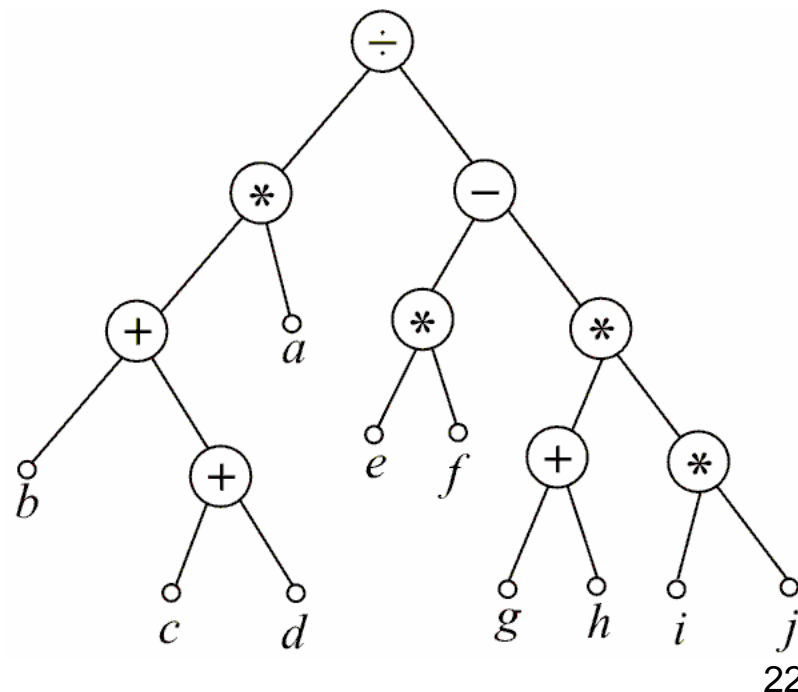


用2叉有序树表示含有2元运算和1元运算的算式: 每个分支点放一个运算符, 其运算对象是以它的儿子为树根的子树所表示的子算式. 规定运算对象的排列顺序, 如被除数、被减数放在左边. 所有的变量和常量都放在树叶上.

例

$$((b+(c+d))*a)\div((e*f)-(g+h)*(i*j))$$

用中序行遍法访问还原算式





**波兰符号法(前缀符号法)**: 按前序行遍法访问存放算式的2叉有序树, 且不加括号.

运算规则: 从右到左每个运算符号对其后面紧邻的两个或一个对象进行运算.

如对上页的算式  $\div * + b + c d a - * e f * + g h * i j$

**逆波兰符号法(后缀符号法)**: 按后序行遍法访问, 且不加括号.

运算规则: 从左到右每个运算符号对其前面紧邻的两个或一个对象进行运算.

如对上页的算式  $b c d + + a * e f * g h + i j * * - \div$



## 主要内容

- 无向树及其性质
- 生成树、最小生成树
- 根树及其分类、最优二叉树、最佳前缀码、波兰符号法、逆波兰符号法

## 基本要求

- 深刻理解无向树的定义及性质
- 熟练地求解无向树
- 准确地求出给定带权连通图的最小生成树
- 理解根树及其分类等概念
- 熟练掌握求最优二叉树及最佳前缀码的方法
- 掌握波兰符号法与逆波兰符号法





1. 无向树  $T$  有  $n_i$  个  $i$  度顶点,  $i=2, 3, \dots, k$ , 其余顶点全是树叶, 求  $T$  的树叶数.

解 用树的性质: 边数  $m=n-1$  ( $n$  为阶数), 及握手定理. 设有  $t$  片树叶,

$$2m = 2\left(\sum_{i=2}^k n_i + t\right) - 2 = \sum_{i=1}^n d(v_i) = \sum_{i=2}^k i n_i + t$$

解得

$$t = \sum_{i=3}^k (i-2)n_i + 2$$



2. 设 $n$ 阶非平凡的无向树 $T$ 中,  $\Delta(T) \geq k$ ,  $k \geq 1$ . 证明 $T$ 至少有 $k$ 片树叶.

证 设 $T$ 有 $s$ 片树叶, 由于 $\Delta(T) \geq k$ , 有

$$2m = 2n - 2 = \sum_{i=1}^n d(v_i) \geq 2(n - s - 1) + k + s$$

解得 $s \geq k$ .



3. 设 $G$ 为 $n$ 阶无向简单图,  $n \geq 5$ , 证明 $G$ 或 $\bar{G}$ 中必含圈.

证一. 反证法. 否则 $G$ 与 $\bar{G}$ 的各连通分支都是树. 设 $G$ 与 $\bar{G}$ 的连通分支的顶点数和边数分别为 $n_i, m_i (1 \leq i \leq s)$ 与 $n'_j, m'_j (1 \leq j \leq t)$ . 于是

$$\frac{n(n-1)}{2} = \sum_{i=1}^s m_i + \sum_{j=1}^t m'_j = \sum_{i=1}^s (n_i - 1) + \sum_{j=1}^t (n'_j - 1) = 2n - (s + t) \leq 2n - 2$$

整理得

$$n^2 - 5n + 4 \leq 0$$

解得

$$1 \leq n \leq 4$$

与 $n \geq 5$ 矛盾



证二. 在 $G$ 与 $\overline{G}$ 中有一个(不妨设为 $G$ )边数

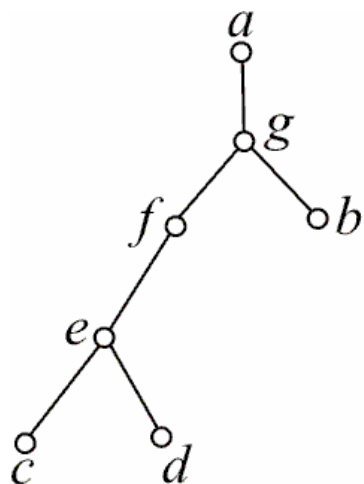
$$m \geq \frac{n(n-1)}{4} \geq n$$

若 $G$ 是森林, 则 $m \leq n-1$ , 矛盾.

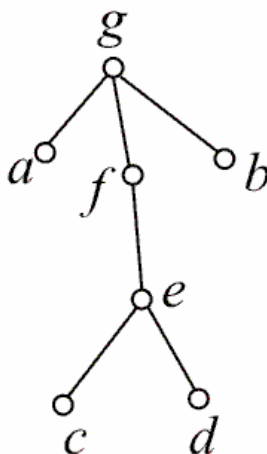


4. 画出基图为图所示无向树的所有非同构的根树

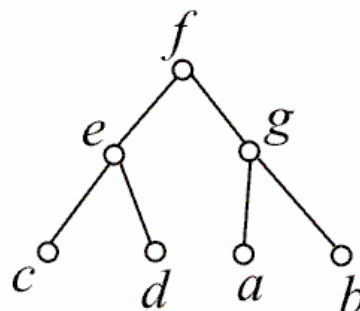
解 以 $a, b, c, d$ 为根的根树同构, 选 $a$ 为根, 如(1);  
以 $e, g$ 为根的根树同构, 取 $g$ 为根, 如(2);  
以 $f$ 为根, 如(3).



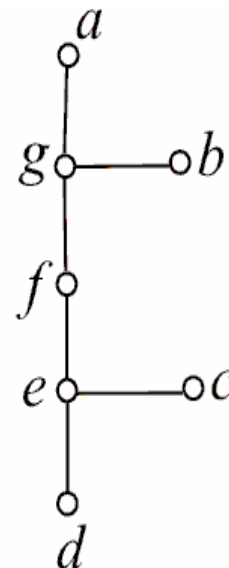
(1)



(2)



(3)





5. 设 $T$ 是正则2叉树,  $T$ 有 $t$ 片树叶, 证明 $T$ 的阶数  $n=2t-1$ .

证一. 利用正则2叉树的定义及树的性质直接证明.

(1)  $n = t + i$  ( $i$ 为分支点数)

(2)  $n = m + 1$  ( $m$ 为 $T$ 的边数)

(3)  $m = 2i$  (正则2叉树定义)

由(2)、(3)得  $i = \frac{n-1}{2}$ , 代入(1)得  $n = 2t-1$ .

证二. 利用握手定理及树的性质证.

$T$ 的树根为2度顶点, 所有内点为3度顶点, 叶为1度顶点, 有

(1)  $2m = 2 + 3(i-1) + t$

(2)  $m + 1 = n = i + t$

由(1)和(2)可解得  $n = 2t-1$ .