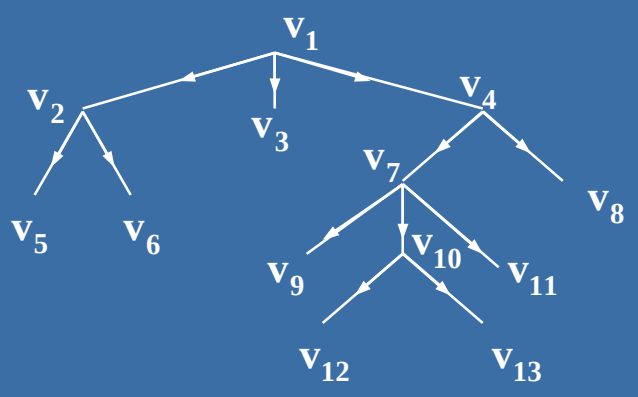
三、根树及其应用

1、根树的定义

定义 11:一个有向图,如果删去每边的方向得到的无向图是一棵树,则称该图为有向树。

定义 12:一棵非平凡的有向树,如果恰有一个点的入度为 0,其余所有点的入度均为 1,则称该树为根树,入度为零的点称为树根,出度为 0 的点称为树叶,出度不为零的点称为内点,内点和树根统称为分支点。



图中是 v_1 树根, v_2, v_4, v_7, v_{10} 是内点,

$v_5, v_6, v_8, v_9, v_{11}, v_{12}, v_{13}$ 是树叶。

习惯把有向树的根画在最上方,边的箭头全指向下,则可以省略全部箭头,树根到一个结点的有向通路的长度称为该点的层数。所有结点的最大层数称为树高。上图树高为 4。

定义 13:在根树中,若 v_i 可达 v_j,且通路长度大于等于 2,则称 v_i 是 v_j 的祖先, v_j 是 v_i 的后代,

若 $<v_i, v_j>$ 是根树的边,则称 v_i 是 v_j 的父亲,

v_j 是 v_i 的儿子,同一结点的儿子称为兄弟。

定义 14:任一结点 v 及其后代导出的子图称为根树的子树。

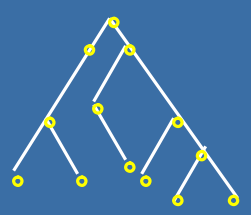
定义 15:在根树中规定了每一层上结点的次序,称为有序树。

2. m 叉树和二叉树

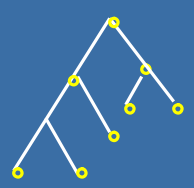
定义 16:在根树中若每个结点的出度均≤m,则称 T 为 m 元树(m 叉树)(即每个点至多有 m 个儿子),若每个分支点的出度恰好等于 m,则称 T 为 m 叉完全树,若 T 的所有树叶的层数均相同,则称 T 正则 m 元树。

若 m 元树是有序的,则称 T 为 m 元有序树,若 m 元完全树是有序的则称 T 为完全 m 元有序树,若 m 元正则树是有序的,则称 T 为 m 元正则有序树。

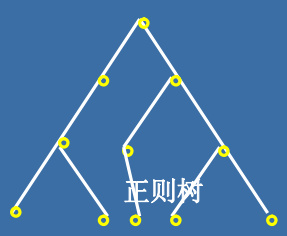
例:



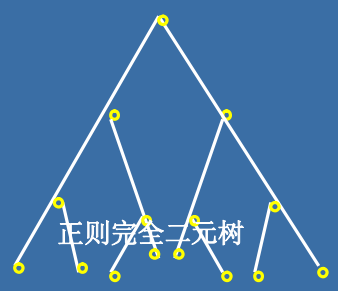
二元树



完全二元树



正则树



正则完全二元树

当 m=2 时,称为二元树,二元有序树的每个结点

至多有两个儿子,其序按左右分,分别为左儿子,

右儿子,任一分支点最多有两棵子树，称为左子树

和右子树。

定理 17:在完全 m 元树 T 中若树叶数为 t,分支点数

为 i 则有 $(m-1)i = t-1$。

证明:T 有 t 个叶,i 个分支点,则有 $t+i$ 个点,$t+i-1$

条边,则所有出度和等于边数∴$m\cdot i=i+t-1$

∴$(m-1)i=t-1$。

定理 18:若二元完全树有 n 个分支点,且各分支点的

层数之和为I,各树叶的层数之和为 E,则 E=I+2n。

证明:对分支点数用数字时纳法,n=1

如图

此时 E=0,I=2,n=1 成立,设 $n=k$ 时或 $n=k+1$ 时,

任选两个为兄弟的树叶,父亲为 v,将 t_1,t_2 去掉,则

$n=k$ 且满足归纳法,$E1=I+2n$ 则在原图中,设 v 的层

数为 S,则 $I=I_1+S$,t_1,t_2 的层数均为 $S+1$,则

$E=E+2(S+1)-S=E_1+S+2$ 由 $E1=I1+2K$ 得

$E=E1+S+2=I1+2K+S+2=(I1+S)+2(K+1)=I+2(K+1)$

结论仍成立。

由定理 17:完全二元树分支点 i 比叶少一个,在完全

正则二元树中，树高是 k，则叶是 2^k 个，分支点是 $2^k - 1$ 个，边是 $2 \times 2^k - 1$ 条边。

四、哈夫曼树和最优树问题

给一组数 $w_1, w_2, \ldots w_t$，不妨假设 $w_1 \leqslant w_2 \leqslant \cdots \leqslant w_t$。若有一棵二叉树，共有 t 片叶，分别带权 $w_1, w_2, \ldots w_t$，称该二叉树为带权二叉树。

定义 17：在带权二叉树 T 中，若带权为 w_i 的树叶的层数 $i=1,\ldots,t$ 为 $L(w_i)$，则称 $w(T) = \sum_{i=1}^{t} w_i L(w_i)$ 为 T 的权，在所有带权为 $w_1, w_2, \ldots w_t$ 的二叉树中，$w(T)$ 最小的那棵树称为最优树。

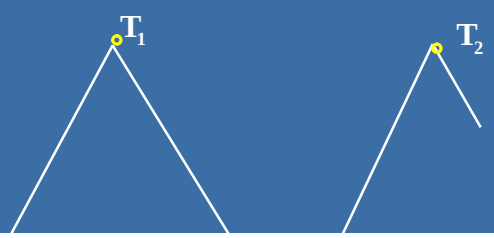
定理 19：设 T 为带权 $w_1 \leqslant w_2 \leqslant \cdots \leqslant w_t$ 的最优树，则

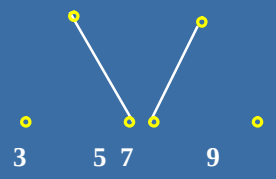
(1) 带权 w_1，w_2 的树叶 v_{w1}，v_{w2} 是兄弟

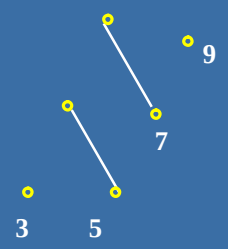
(2) 以树叶 v_{w1}，v_{w2} 为儿子的分枝点的通路最长

证明：略，即权越小的叶要挂在路最长的分支点，权越大挂的叶要挂在路短的分支点上，显然最优树一定是完全二叉树。

例：







$w(T_1) = (3+5+7+9) \times 2 = 48$　　$w(T_2) = 3(5+3) + 2 \times 7 + 9 \times 1 = 47$

定理 20：设 T 为带权 $w_1 \leqslant w_2 \leqslant \cdots \leqslant w_t$ 的最优树，

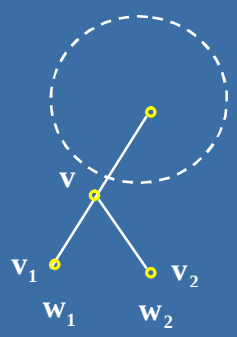
若将以带权 w_1 和 w_2 的树叶为儿子的分枝改为带权

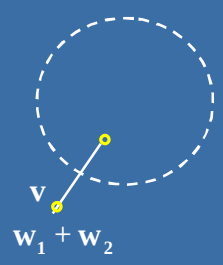
w_1+w_2 的树叶，得到的新树 T'，则 T' 是带权为，

$w_1+w_2, w_3, w_4, \cdots, w_t$ 的最优树。

证明：

根据假设 v 的两个儿子的层数，比 v 的层数多 1。





$\therefore w(T) = w(T') + w_1 + w_2$，若 T' 不是带权 $w_1 + w_2, w_3, \ldots, w_t$

的最优树，则必存在一个带以上权的最优树 T''，

使 $w(T') > w(T'')$，在 T'' 中带权 $w_1 + w_2$ 的树叶 v,生成

两个儿子,各带权 w_1 和 w_2,得一个新树 $\bar{T}$,它是带权

$w_1, w_2, \ldots w_t$ 的树，$w(\bar{T}) = w(T'') + w_1 + w_2$ 由此得到

$w(T) = w(T') + w_1 + w_2 > w(T'') + w_1 + w_2$ 这与 T 是带权

$w_1, w_2, \ldots w_t$ 的最优树矛盾。

此定理就导出了最优树的哈夫曼算法，由该算法得到的树称为哈夫曼树，即最优树。

哈夫曼算法：给定树 $w_1, w_2, \ldots w_t$ 求最优树

1. 给初始权集 $S=\{w_1, w_2, \ldots w_t\}$ t 个叶 v_i 带权 w_i，$i = 1, \ldots, t$，

2. 在 S 中找出两个权最小的数不妨记作 w_1, w_2，用父结点 v 将两个带权的儿子 v_1, v_2 连结起来，形成一个新的子树并把该子树看作一个结点 v，带权 w_1+w_2。

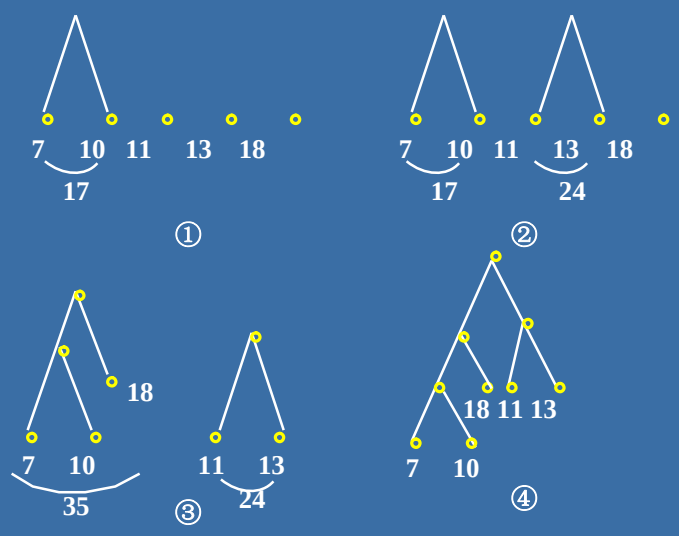
3. 置权集 $S := (S-\{w_1, w_2\} \cup \{w_1+w_2\}$

4. 检查 S 中是否只有一个元素？

是就停止，否则转入 2

例：求带权 7，10，11，13，18 的最优树

解：



此时，$w(T) = 7 \times 3 + 10 \times 3 + 18 \times 2 + 11 \times 2 + 13 \times 2 = 133$