7-8 根树及其应用

前面我们讨论的树，都是一个无向图，下面我们讨论有向图的树。

【定义7-8.1】有向树

如果一个有向图在不考虑边的方向时是一棵树，那么，这个有向图称为有向树。

7-8 根树及其应用

【定义7-8.2】根树

一棵有向树，如果有一个结点的入度为0，其余所有结点的入度都为1，则称为根树。入度为0的结点称为根，出度为0的结点称为叶，出度不为0的结点称为分支点或内点。

7-8 根树及其应用

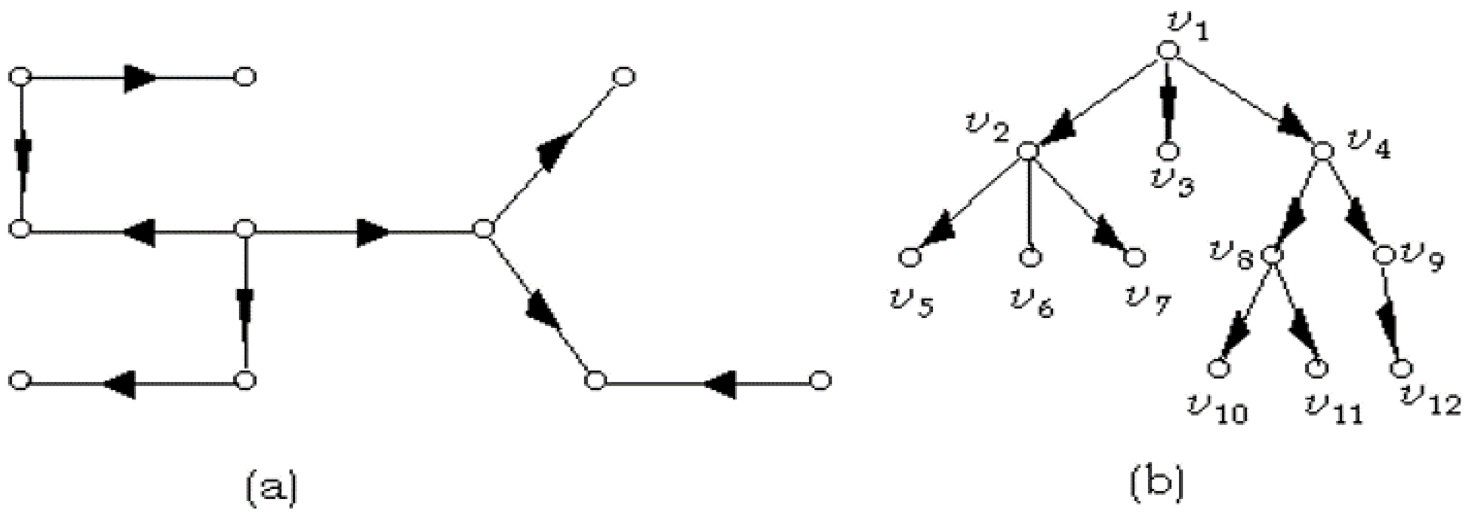


图7-8.1 有向树和根树

7-8 根树及其应用

例如图7-8.1(b)表示的是一棵根树，其中v_1为根，v_2，v_4，v_8，v_9为分枝点，其余结点为叶。

在根树中，任意一个结点v的层次，就是从根到该结点的单向通路的长度。

在图7-8.1(b)中，有三个结点的层次为1，有五个结点的层次为2，有三个结点的层次为3。

从根树的结构可以看出，树中每一个结点可以看作是原来树中的某一棵子树的根，由此可知，根树亦可递归定义为：

7-8 根树及其应用

【定义7-8.3】 根树包括一个或多个结点，这些结点中某一个称为根，其它所有结点被分成有限个子根树。

这个定义把n个结点的根树用结点数少于n的根树来定义，最后得到每一棵都是一个结点的根树，它就是原来那棵树的树叶。

对于一棵根树，如果用图形来表示，可以有树根在下或树根在上的两种画法。

7-8 根树及其应用

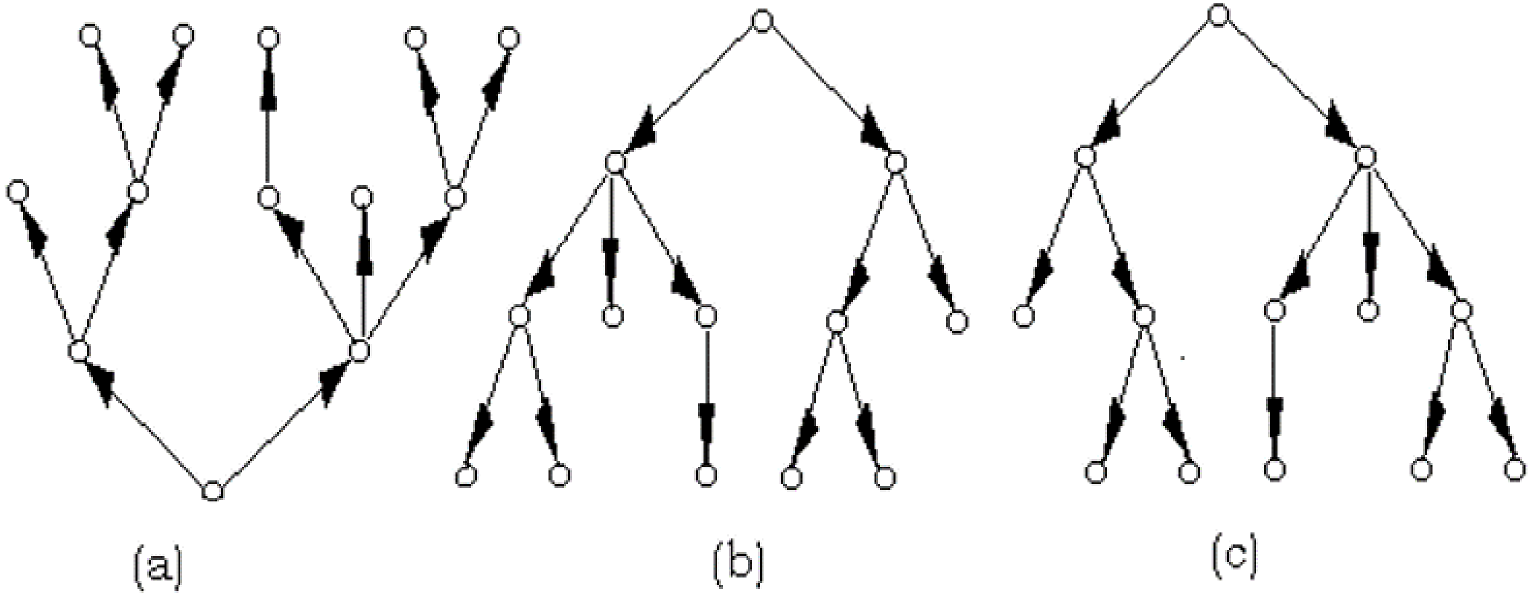


图7-8.2 根树的两种画法

7-8 根树及其应用

图7-8.2(a)是根树自然表示法，即树从它的树根向上生长。图7-8.2(b)和(c)都是由树根往下生长，它们是同构图，其差别仅在于每一层上的结点从左到右出现的次序不同，为此今后要用明确的方式，指明根树中的结点或边的次序，这种树称为有序树。

7-8 根树及其应用

设a是根树中的一个分枝点，假若从a到b有一条边，则结点a称为结点b的“父亲”，而结点b称为结点a的“儿子”。假若从a到c有一条单向通路，称结点a为结点c的“祖先”，而结点c称为结点a的“后裔”。同一个分枝点的“儿子”称为“兄弟”。

m叉树是一种特殊的根树，在m=2时，称为二叉树，它在计算机科学中有着广泛的应用。

7-8 根树及其应用

【定义7-8.4】 在根树中，若每一个结点的出度小于或等于m，则这棵树称为m叉树。如果每一个结点的出度恰好等于m或零，则这棵树称为正则m叉树。若所有的树叶层次相同，则这棵树称为完全m叉树，若m=2时，称为二叉树。

有许多实际问题可用二叉树或m叉树来表示。

7-8 根树及其应用

例如M和*E*两人进行网球比赛，如果一人连胜两盘或共胜三盘就获胜，比赛结束。共有多少中获胜的方式？

图**7-8.3**表示了比赛可能进行的各种情况，它有十片树叶，从根到树叶的每一条路对应比赛可能发生的一种情况，即：**MM**，**MEMM**，**MEMEM**，**MEMEE**，**MEE**，**EMM**，**EMEMM**，**EMEME**，**EMEE**，**EE**。

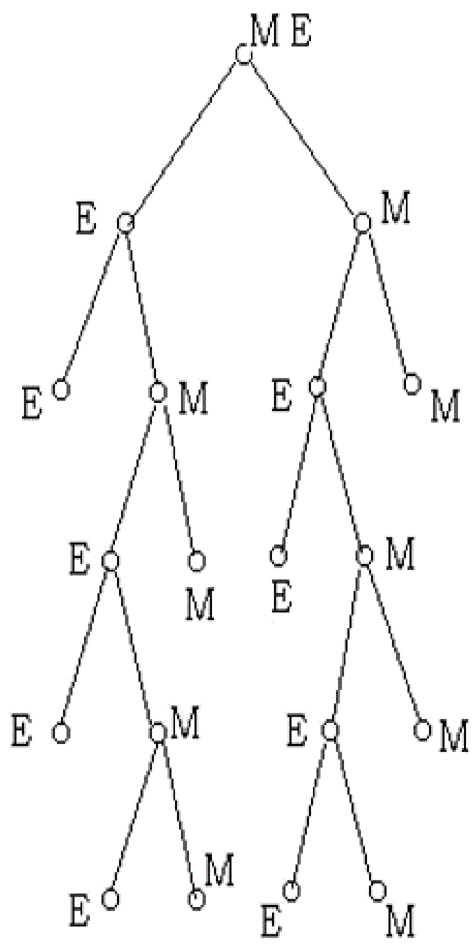


图7-8.3

7-8 根树及其应用

我们要指出，任何一棵有序树都可以改写为对应的二叉树。如图**7-8.4(a)**中的**m**叉树可用下述方法改为二叉树：

7-8 根树及其应用

⑴除了最左边的分枝点外，删去所有从每一结点长出的分枝。在同一层次中，兄弟结点间用从左到右的有向边连接，如图**7-8.4(b)**所示。

⑵选定二叉树的左儿子和右儿子如下：直接处于给定结点下面的结点，作为左儿子，对于同一水平线上给定结点右邻结点，作为右儿子，以此类推，如图**7-8.4(c)**所示。

7-8 根树及其应用

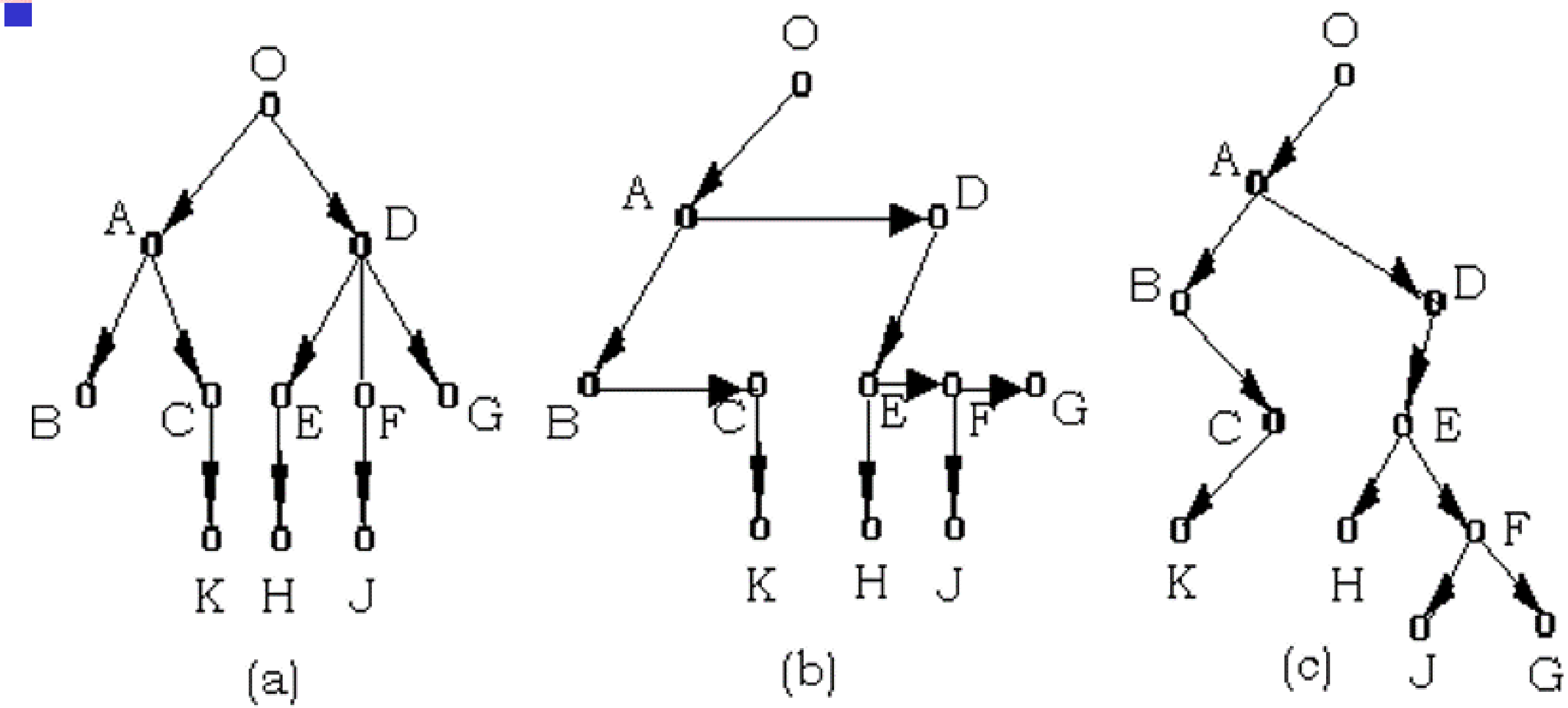
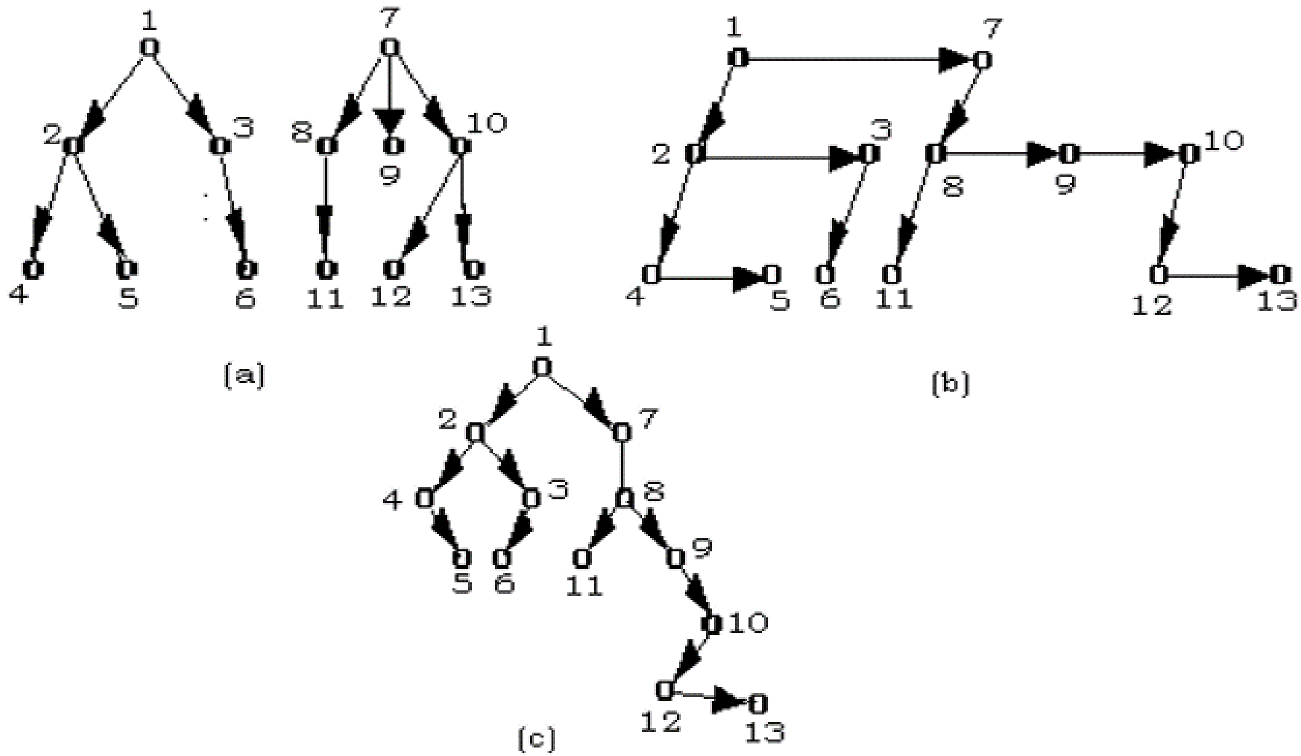


图7-8.4 m叉树改写为二叉树

7-8 根树及其应用

用二叉树表示有序根树的方法，可以推广到有序森林上去,如图所示。



7-8 根树及其应用

在树的实际应用中，我们经常研究正则m叉树。

定理7-8.1 设有正则m叉树，其树叶数为t，分枝点数为i，则$(m-1)i=t-1$。

证明 若把m叉树当作是每局有m位选手参加比赛的单淘汰赛计划表，树叶数为t表示参加比赛的选手数，分枝点数为 i 表示比赛的局数，

因为每局比赛将淘汰$(m-1)$位选手，比赛的结果共淘汰$(m-1)i$位选手，最后剩下一个冠军，

因此$(m-1)i+1=t$

即$(m-1)i=t-1$。

7-8 根树及其应用

例1. 设有28盏灯，拟共用一个电源插座，问需用多少块具有四种插座的接线板。

解 将四叉树每个分枝点看作是具有四插座的接线板，树叶看作电灯，则$(4-1)i=28-1$, $i=9$，所以需要九块具有四插座的接线板。

7-8 根树及其应用

例2 设有一台计算机，它有一条加法指令，可计算三个数的和，如果要计算九个数的和，至少要执行几次加法命令。

解 若把九个数看作是正则三叉树的九片树叶，则有 $(3-1)i=9-1$, $i=4$，所以，需要执行四次加法指令。

7-8 根树及其应用

在计算机的应用中，还常常考虑二叉树的通路长度问题。

【定义7-8.5 】 在根树中，一个结点的通路长度，就是从树根到此结点的通路中的边数。我们把分枝点的通路长度称作内部通路长度，树叶的通路长度称作外部通路长度。

7-8 根树及其应用

定理7-8.2 若正则二叉树有n个分枝点，且内部通路长度总和为I，外部通路长度总和为E，则

$E=I+2n$

证明 对分枝点数目n进行归纳。

当$n=1$时，$E=2$，$I=0$，故$E=I+2n$成立。

假设$n=k-1$时成立，即$E'=I'+2(k-1)$。

当$n=k$时，若删除去一个具有两个树叶的分枝点v的两个树叶，得到新树T'。设该分枝点与根的通路长度为l。将T'与原树比较，它减少了两片长度为$l+1$的树叶和一个长度为l的分枝点，但增加了1个长为l的树叶。因为T'有$(k-1)$个分枝点，故$E'=I'+2(k-1)$。

但在原树中，有$E=E'+2(l+1)-l=E'+l+2$, $I=I'+l$，代入上式得$E-l-2=I-l+2(k-1)$，即$E=I+2k$。

7-8 根树及其应用

二叉树的一个重要应用就是最优树问题。

给定一组权w_1，w_2，…，w_t，不妨设$w_1 \leqslant w_2 \leqslant \cdots \leqslant w_t$。设有一棵二叉树，共有$t$片树叶，分别带权$w_1$，$w_2$，…，$w_t$，该二叉树称为带权二叉树。

7-8 根树及其应用

[定义7-8.6] 在带权二叉树T中，若带权为w_i的树叶，其通路长度为$L(w_i)$,把

$$w(T) = \sum_{i=1}^{t} w_i L(w_i)$$

称为该**带权二叉树的权**，所有带权$w_1, w_2, \ldots, w_t$的二叉树中，$w(T)$最小的那棵树，称为**最优树**。

7-8 根树及其应用

假若给定一组权w_1，w_2，…，w_t，为了找最优树，我们先证明下面定理：

定理7-8.3 设T为带权$w_1 \leqslant w_2 \leqslant \cdots \leqslant w_t$的最优树，则

a）带权为w_1，w_2的树叶是兄弟。

b) 以树叶w_1，w_2为儿子的分枝点，其通路长度最长。

7-8 根树及其应用

证明 设在带权w_1，w_2，…，w_t的最优树中，v是通路长度最长的分枝点，v的儿子分别为w_x和w_y，故有

$$L(w_x)\geqslant L(w_1)$$

$$L(w_y)\geqslant L(w_2)$$

若有$L(w_x)>L(w_1)$，将w_x与w_1对调，得到新树T'，则

$w(T')-w(T)=(L(w_x)\cdot w_1+L(w_1)\cdot w_x)-(L(w_x)\cdot w_x+L(w_1)\cdot w_1)$

$=L(w_x)(w_1-w_x)+L(w_1)(w_x-w_1)=(w_x-w_1)(L(w_1)-L(w_x))<0$。

即$w(T')<w(T)$与T是最优树的假定矛盾。故$L(w_x)=L(w_1)$。

同理可证$L(w_x)=L(w_2)$。因此

$L(w_1)=L(w_2)=L(w_x)=L(w_y)$

分别将w_1，w_2与w_x，w_y对调得到一棵最优树，其中带权w_1和w_2的树叶是兄弟。

7-8 根树及其应用

定理7-8.4 设T为带权$w_1 \leqslant w_2 \leqslant \cdots \leqslant w_t$的最优树，若将以带权$w_1$，$w_2$的树叶为儿子的分枝点改为带权$w_1+w_2$的树叶，得到一棵新树$T'$，则$T'$也是最优树。

证明 根据题设，有

$w(T)=w(T')+w_1+w_2$。

若T'不是最优树，则必有另外一棵带权为w_1+w_2，w_3，…，w_t的最优树T''。对T''中带权为w_1+w_2的树叶v_{w1+w2}生成两个儿子，得到树$\hat{T}$，则

$w(\hat{T})=w(T'')+w_1+w_2$。

7-8 根树及其应用

因为T''是带权为w_1+w_2，w_3，…，w_t的最优树树，故

$w(T'')\leq w(T')$。

如果$w(T'')<w(T')$，则$w(\ )<w(T)$，与T是带权为w_1，w_2，…，w_t最优树矛盾，因此

$w(T'')=w(T')$

T'是一棵带权为w_1+w_2，w_3，…，w_t的最优树。

7-8 根树及其应用

根据上面两个定理，要画一棵带有t个权的最优树，可简化为画一棵带有t-1个权的最优树，而又可简化为画一棵带t-2个权的最优树，依此类推。具体的做法是：首先找出两个最小w值，设为w_1和w_2，然后对t-1个权w_1+w_2，w_3，…，w_t求作一棵最优树，并且将这棵最优树的结点v_{w1+w2}分叉生成两个儿子v_1和v_2，依此类推。此称为Huffman算法。

7-8 根树及其应用

例3. 设一组权2，3，5，7，11，13，17，19，23，29，31，37，41。求相应的最优树。

解：见下页图

7-8 根树及其应用

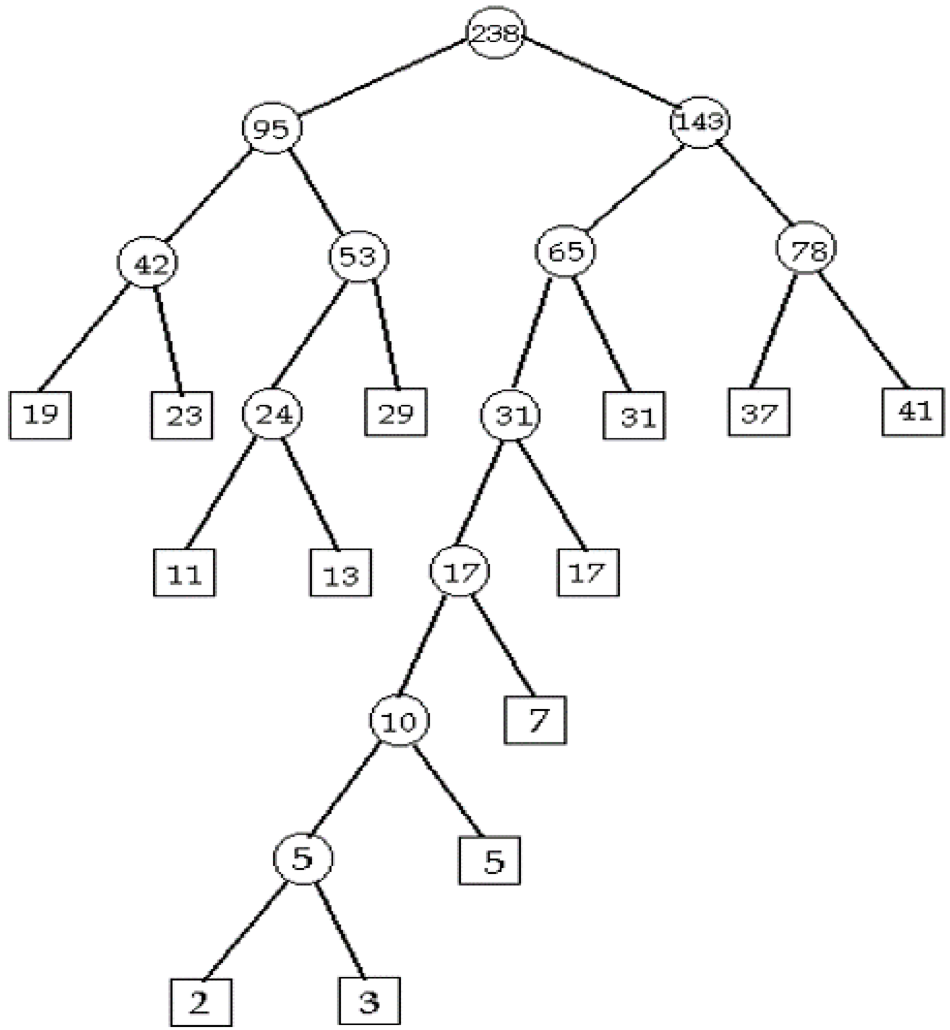


图7-8.7 最优二叉树

7-8 根树及其应用

二叉树的另一个应用，就是前缀码问题。

我们知道，在远距离通讯中，常常用0和1的字符串作为英文字母传送信息，因为英文字母共有26个，故用不等长的二进制序列表示26个英文字母时由于长度为1的序列有2个，长度为2的二进制序列有2^2个，长度为3的二进制序列有2^3个，依此类推，我们有

$$2+2^2+2^3+\cdots+2^i \geqslant 26$$

$$2^{i+1}-2 \geqslant 26,\ i \geqslant 4$$

7-8 根树及其应用

因此，用长度不超过四的二进制序列就可表达26个不同英文字母。但是由于字母使用的频繁程度不同，为了减少信息量，人们希望用较短的序列表示频繁使用的字母。当使用不同长度的序列表示字母时，我们要考虑的另一个问题是如何对接收的字符串进行译码？

7-8 根树及其应用

定义7-8.7 给定一个序列的集合，若没有一个序列是另一个序列的前缀，该序列集合称为前缀码。

例如{000，001，01，10}是前缀码，

而{1，0001，000}就不是前缀码。

7-8 根树及其应用

定理7-8.5 任何一棵二叉树的树叶可对应一个前缀码。

证明 给定一棵二叉树，从每一个分枝点引出两条边，对左侧边标以0，对右侧边标以1，则每片树叶可以标定一个0和1的序列，它是由树根到这片树叶的通路上各边标号所组成的序列，显然，没有一片树叶的标定序列是另一片树叶的标定序列的前缀，因此，任何一棵二叉树的树叶可对应一个前缀码。

7-8 根树及其应用

定理7-8.6 任何一个前缀码都对应一棵二叉树。

证明 设给定一个前缀码，h表示前缀码中最长序列的长度。我们画出一棵高度为h的完全正则二叉树，并给每一分枝点射出的两条边标以0和1，这样，每个结点可以标定一个二进制序列，它是从树根到该结点通路上各边的标号所确定，因此，对长度不超过h的每一二进制序列必对应一个结点。对应于前缀码中的每一序列的结点，给予一个标记，并将标记结点的所有后裔和射出的边全部删去，这样得到一棵二叉树，再删去其中从根到标记结点路径以外的所有分支点和树叶，得到一棵新的二叉树，它的树叶就对应给定的前缀码。

7-8 根树及其应用

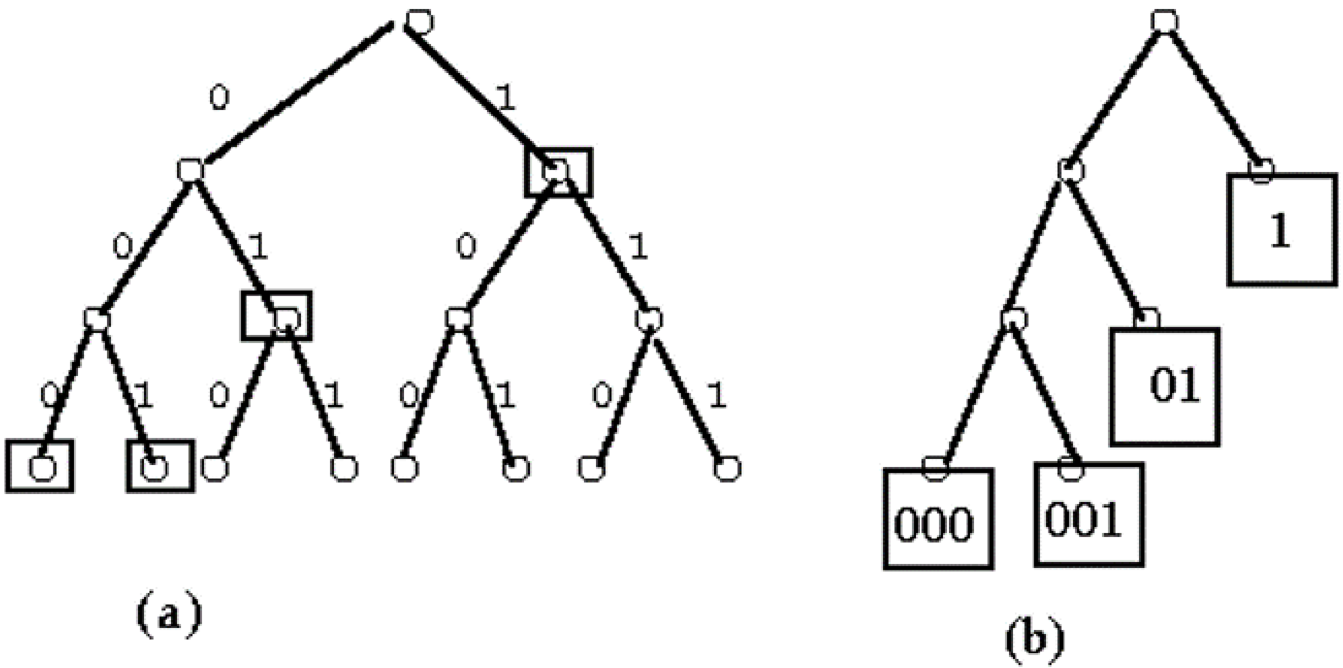


图7-8.8 二叉树对应前缀码

7-8 根树及其应用

例如，图7-8.8给出了与前缀码{000，001，1}对应二叉树，其中图(a)是高度为3的完全正则二叉树，对应前缀码中序列的结点用方框标记，图(b)是对应的二叉树。

通过前缀码和二叉树的对应关系，我们可知，如果给定前缀码对应的二叉树是正则二叉树，则此前缀码可进行译码。

7-8 根树及其应用

例如图7-8.8(b)中所对应的前缀码{000,001，01，1}，

可对任意二进制序列进行译码。

设有二进制序列

000100110111101001

可译为000,1,001,1,01,1，1,01,001。

如果被译的信息最后部分不能译前缀码中的序列，

可约定添加0或1，直至能够译出为止。

最佳前缀码

设要传输的电文中含有t个字符, 字符a_i出现的频率为p_i, 它的编码的长度为l_i, 那么100个字符的电文的编码的期望长度是$100\sum_{i=1}^{t} l_i p_i$. 称编码期望长度最小的2元前缀码为**最佳2元前缀码**.

在用2叉树产生2元前缀码时, 每个二进制串的长度等于它所在树叶的深度, 因而权为$100p_1$, $100p_2$, …, $100p_t$的最优2叉树产生的2元前缀码是最佳2元前缀码. 于是, 给定字符出现的频率, 可以用Huffman算法产生最佳2元前缀码.

实例

例 在通信中，设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%

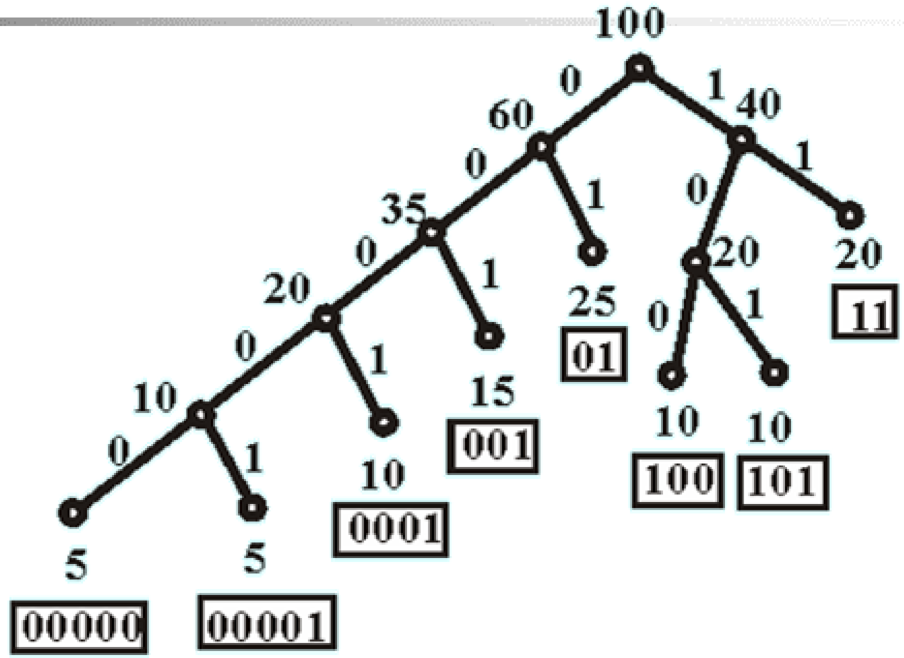
采用2元前缀码, 求传输数字最少的2元前缀码，并求传输 $10^n(n\geq 2)$个按上述比例出现的八进制数字需要多少个二进制数字？若用等长的 (长为3) 的码字传输需要多少个二进制数字?

解 用Huffman算法求以频率(乘以100)为权的最优2叉树. 这里 $w_1=5, w_2=5, w_3=10, w_4=10, w_5=10, w_6=15, w_7=20, w_8=25$.

例(续)

编码:

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



传100个按比例出现的八进制数字所需二进制数字的个数为 $W(T)$=285.

传$10^n(n\geq 2)$个所用二进制数字的个数为2.85×10^n, 而用等长码(长为3)需要用 3×10^n个数字.

39

行遍2叉有序树

行遍(周游)根树T：对T的每个顶点访问且仅访问一次.

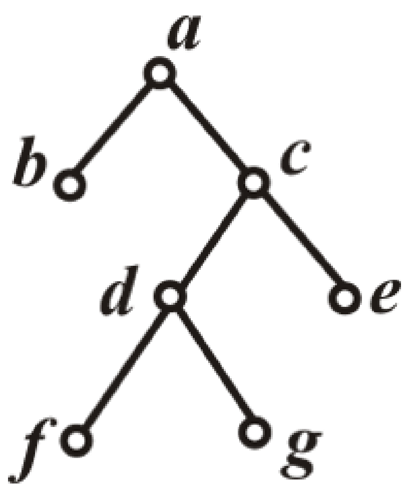
行遍2叉有序树的方式：

① 中序行遍法: 左子树、根、右子树

② 前序行遍法: 根、左子树、右子树

③ 后序行遍法: 左子树、右子树、根

当不是正则树时, 左子树或右子树可缺省



例如, 中序行遍: $b\ \underline{a}\ (f\ \underline{d}\ g)\ \underline{c}\ e$

前序行遍: $\underline{a}\ b\ (\underline{c}\ (\underline{d}\ f\ g)\ e)$

后序行遍: $b\ ((f\ g\ \underline{d})\ e\ \underline{c})\ \underline{a}$

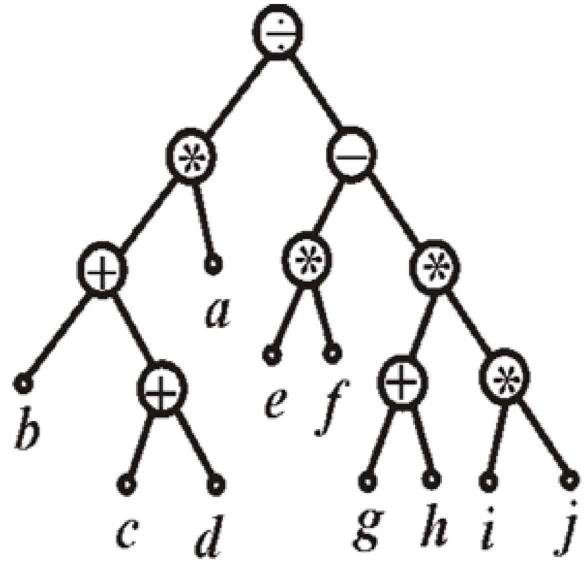
40

用2叉有序树表示算式

每一个分支点放一个运算符. 二元运算符所在的分支点有2个儿子, 运算对象是以这2个儿子为根的根子树表示的子表达式, 并规定被减数和被除数放在左子树上; 一元运算符所在的分支点只有一个儿子, 运算对象是以这个儿子为根的根子树表示的子表达式.数字和变量放在树叶上.

实例

例1 表示$((b+(c+d))*a)\div((e*f)-(g+h)*(i*j))$的2叉有序树



中序行遍:

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

前序行遍:

$\div(*(+b(+cd))a)(-(*ef)(*(+gh)(*ij)))$

后序行遍:

$((b(cd+)+)a*)((ef*)((gh+)(ij*)*)-)\div$

注:中序行遍的结果是原式

42

波兰符号法

波兰符号法(前缀符号法): 按前序行遍法访问表示算式的2叉有序树, 并舍去所有括号.

例1(续) $\div * + b + c d a - * e f * + g h * i j$

计算方法: 从左到右, 每个运算符号对它后面紧邻的2个(或1个)数进行运算.

例1(续) 设a=3, b=1, c=d=2, e=f=3, g=i=1, h=j=2.

$\div * + 1 \underline{+ 2 2} 3 - * 3 3 * + 1 2 * 1 2$, $\div * \underline{+ 1 4} 3 - * 3 3 * + 1 2 * 1 2$

$\div \underline{* 5 3} - * 3 3 * + 1 2 * 1 2$, $\div (15) - \underline{* 3 3} * + 1 2 * 1 2$

$\div (15) - 9 * \underline{+ 1 2} * 1 2$, $\div (15) - 9 * 3 \underline{* 1 2}$

$\div (15) - 9 \underline{* 3 2}$, $\div (15) \underline{- 9 6}$, $\underline{\div (15) 3}$, 5

43

逆波兰符号法

逆波兰符号法(后缀符号法): 按后序行遍法访问表示算式的2叉有序树, 并舍去所有括号.

例1(续)　$bcd++a*ef*gh+ij**-\div$

计算方法: 从右到左, 每个运算符号对它前面紧邻的2个(或1个)数进行运算.

例1(续)　$122++3*33*12+\underline{12*}*-\div$

$122++3*33*\underline{12+}2*-\div$,　$122++3*33*\underline{32*}-\div$

$122++3*\underline{33*}6-\div$,　$122++3*\underline{96-}\div$

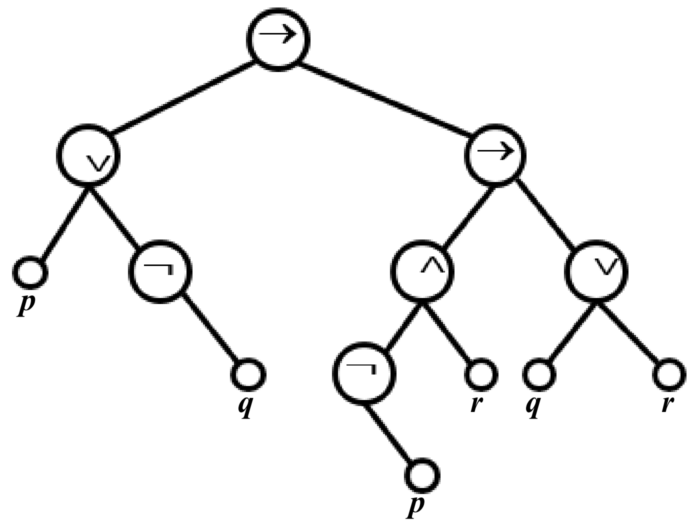
$1\underline{22+}+3*3\div$,　$\underline{14+}3*3\div$,　$\underline{53*}3\div$,　$\underline{(15)3\div}$,　5

实例

例2 用2叉有序树表示下述命题公式, 并写出它的波兰符号法和逆波兰符号法表达式.

$$(p\vee\neg q)\to((\neg p\wedge r)\to(q\vee r))$$

解



波兰符号法表达式

$$\to\vee p\neg q\to\wedge\neg pr\vee qr$$

逆波兰符号法表达式

$$pq\neg\vee p\neg r\wedge qr\vee\to\to$$

注: 当一元运算符在运算对象前面时, 应画成右儿子.