16.3 根树及其应用

定义16.6 T是有向树（基图为无向树的有向图）

(1) T 为根树——T 中一个顶点入度为0，其余顶点入度均为1.

(2) 树根——入度为0的顶点

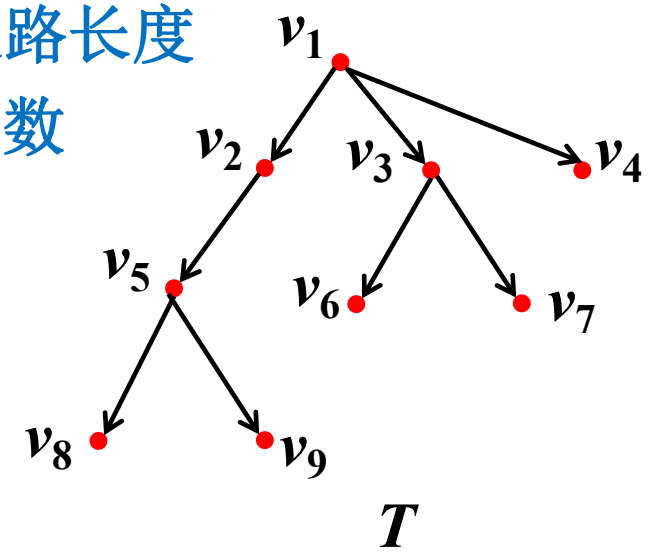
(3) 树叶——入度为1，出度为0的顶点

(4) 内点——入度为1，出度不为0的顶点

(5) 分支点——树根与内点的总称

(6) 顶点v的层数——从树根到v的通路长度

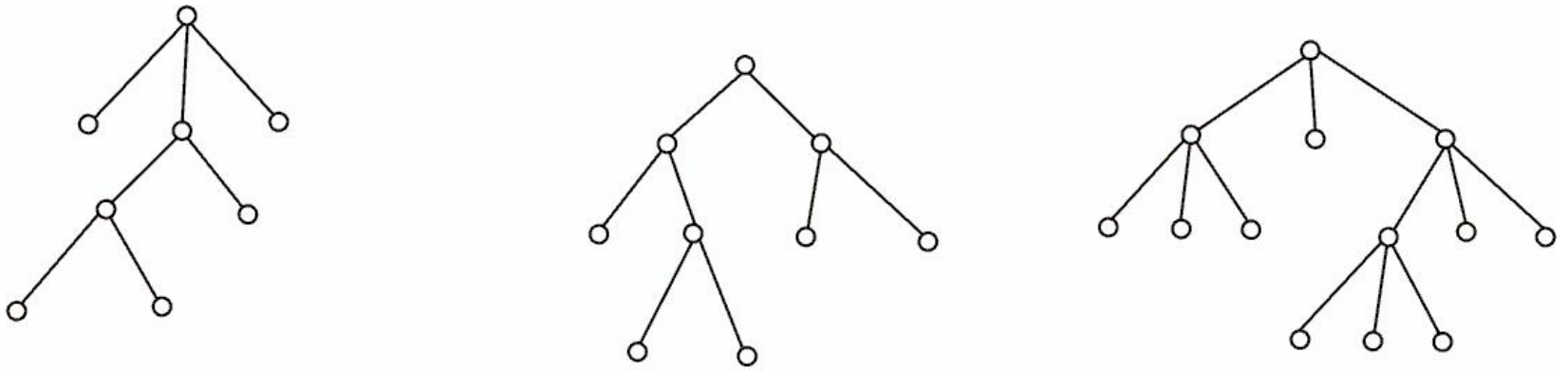
(7) 树高——T 中所有顶点的最大层数



T

根树实例

根树的画法——树根放上方，有向边的方向向下或斜下方，并省去各边上的箭头.



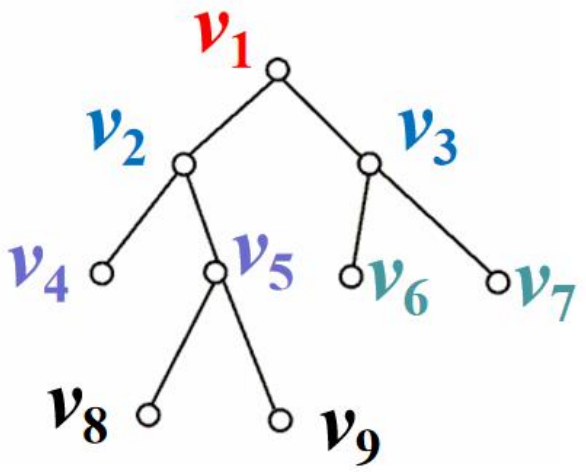
家族树与根子树

定义16.7 设T为一颗非平凡的根树，$\forall v_i, v_j \in V$

(1) 若$v_i \to v_j$，称v_i为v_j的祖先，v_j为v_i的后代

(2) 若$<v_i, v_j> \in E(T)$，称v_i为v_j的父亲，v_j为v_i的儿子

(3) 若v_i，v_j的父亲相同，称v_i为v_j兄弟



根树的分类

(1) T 为有序树——同层数上顶点标定次序的根树（左右顺序）

(2) 根据根树T中每个分支点的儿子数以及是否有序，将根树分为下列各类：

① r 叉树——每个分支点至多有r 个儿子

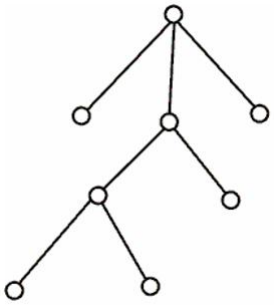
② r 叉有序树——有序的r 叉树

③ r 叉正则树——每个分支点恰有r 个儿子

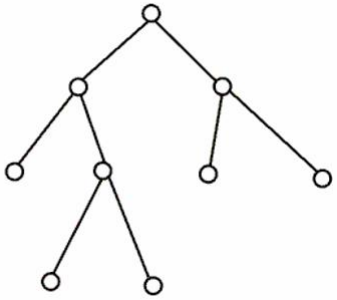
④ r 叉正则有序树——有序的r 叉正则树

⑤ r 叉完全正则树——每个树叶的层数均为树高的r叉正则树

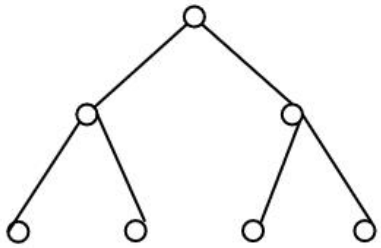
⑥ r 叉完全正则有序树——有序的r 叉完全正则树



3叉树



2叉正则树



2叉完全正则树

2叉正则树的特点

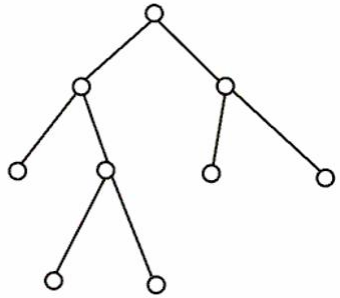
在所有的r叉树中，最常用的是2叉树.

2叉正则树的特点：

树根：度数为2（入度为0，出度为2）；

内点：度数为3（入度为1，出度为2）

树叶：度数为1（入度为1，出度为0）



2叉正则树

例5 已知2叉正则树T有7片树叶，则T有多少个顶点？

设T有n个顶点，由握手定理可得：

$$7+2+3(n-7-1)=2(n-1)$$

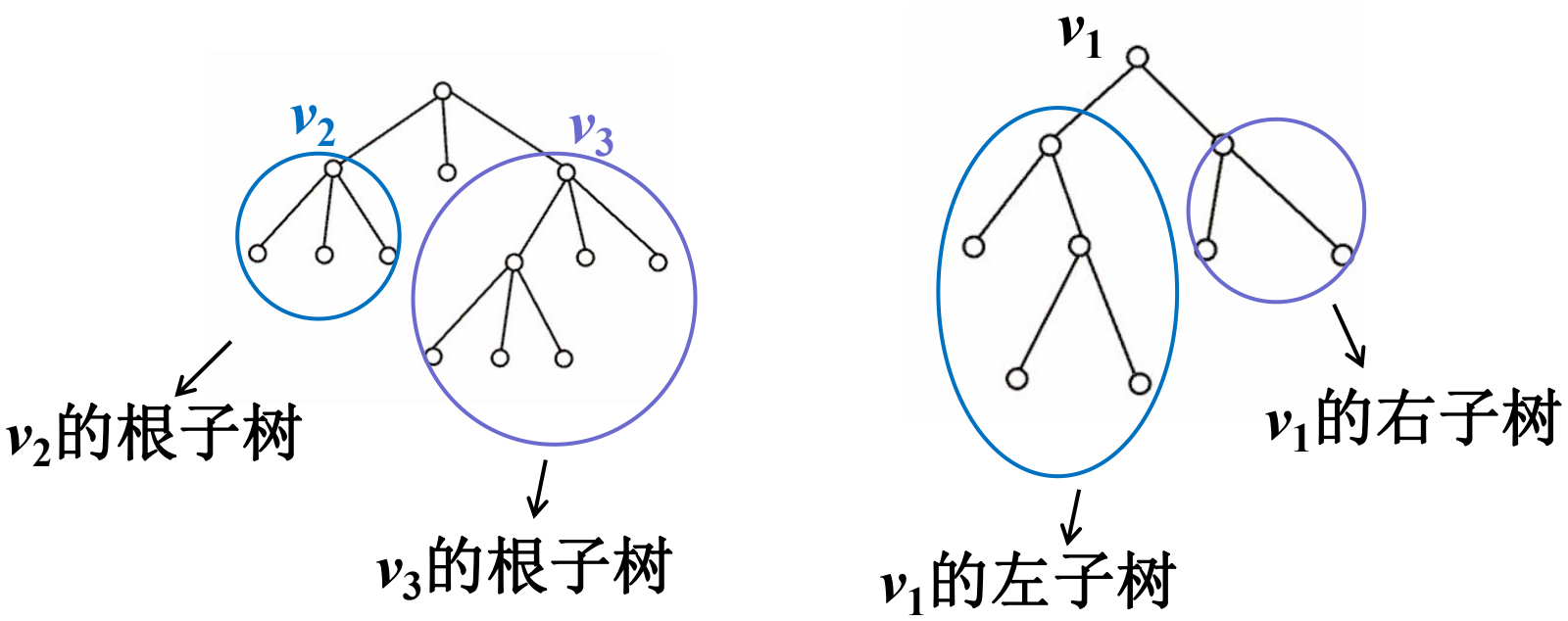
解得：$n=13$；

思考：若2叉正则树T有x片树叶，则T有多少个顶点？

根子树

定义16.8 设v为根树T中任意一顶点，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作该分支点的左子树和右子树.



最优二叉树

定义16.9 设2叉树T有t片树叶$v_1, v_2, \ldots, v_t$，权分别为$w_1, w_2, \ldots, w_t$，称$W(T)=\sum_{i=1}^{t} w_i l(v_i)$为$T$的权，其中$l(v_i)$是$v_i$的层数. 在所有有$t$片树叶，带权$w_1, w_2, \ldots, w_t$的2叉树中，权最小的2叉树称为**最优2叉树**.

图 16.7 所示的 3 棵树 T_1, T_2, T_3 都是带权为 2,2,3,3,5 的 2 叉树. 它们的权分别为

$$W(T_1)=2\times2+2\times2+3\times3+5\times3+3\times2=38$$

$$W(T_2)=3\times4+5\times4+3\times3+2\times2+2\times1=47$$

$$W(T_3)=3\times3+3\times3+5\times2+2\times2+2\times2=36$$

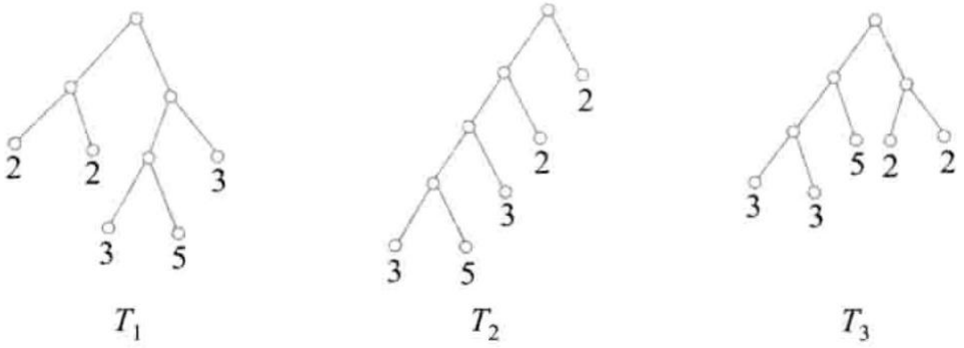


图 16.7

最优二叉树

求最优2叉树的算法—— **Huffman算法**

给定实数$w_1, w_2, \ldots, w_t$，且$w_1 \le w_2 \le \ldots \le w_t$.

(1) 连接权为w_1, w_2的两片树叶，得一个分支点，其权为w_1+w_2.

(2) 在$w_1+w_2, w_3, \ldots, w_t$中选出两个最小的权，连接它们对应的顶点(不一定是树叶)，得新分支点及所带的权.

(3) 重复(2)，直到形成 t–1个分支点，t片树叶为止.

例6 求带权为2, 2, 3, 3, 5的最优2叉树. $W(T)=34$

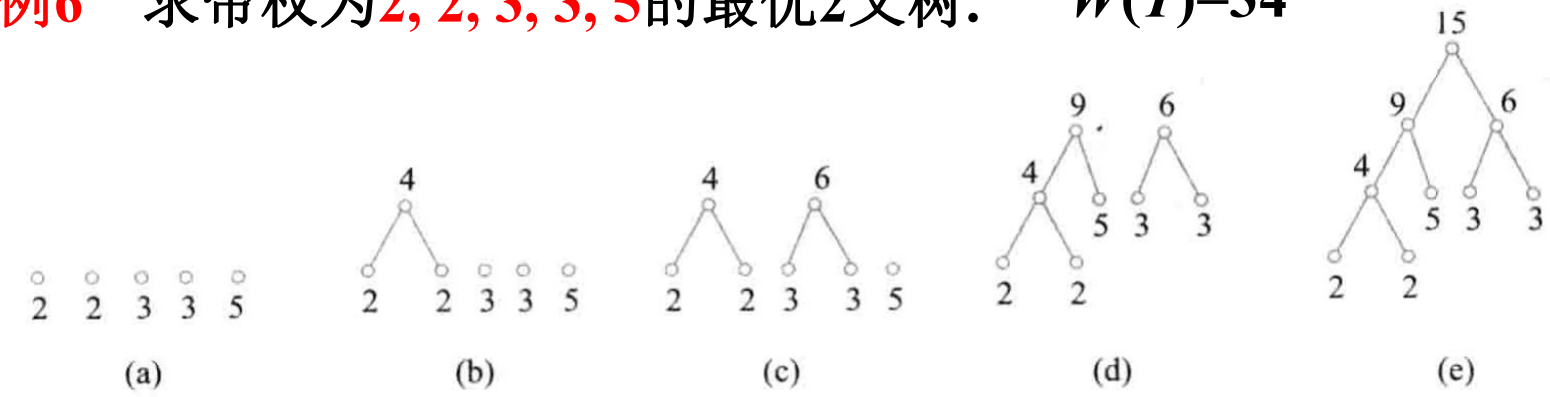
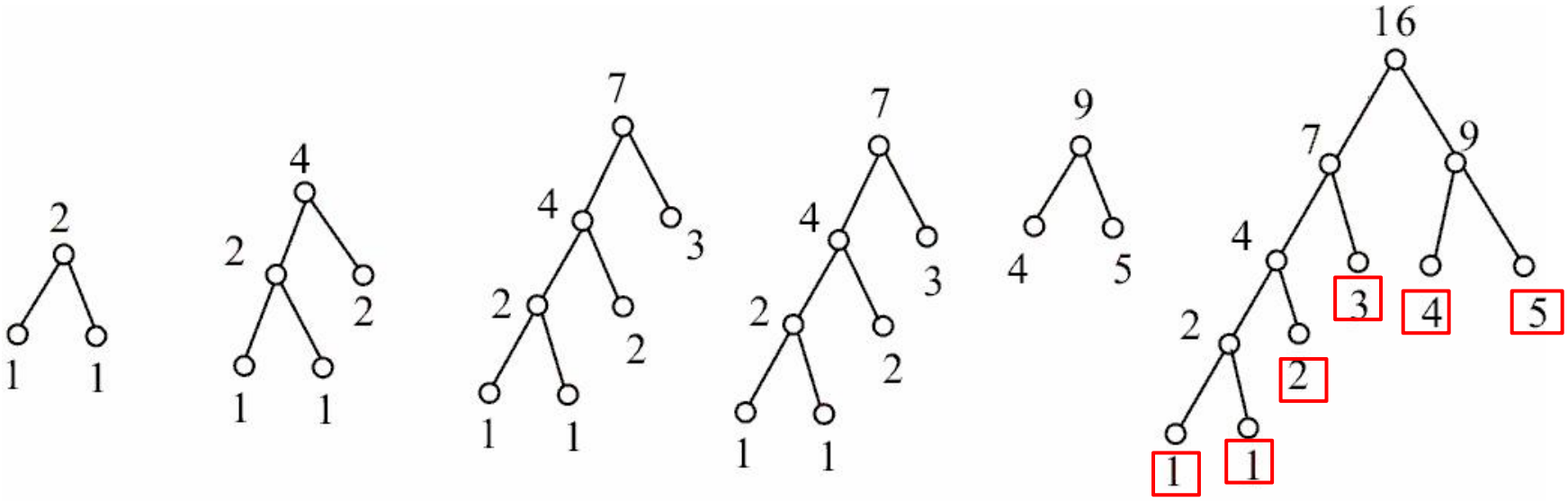


图 16. 8

例 7　求带权为1, 1, 2, 3, 4, 5的最优2叉树.

解题过程由图给出：

$W(T)=1\times4+1\times4+2\times3+3\times2+4\times2+5\times2=38$



二进制编码

在通信中，常用二进制编码表示符号. 例如，可以用长为2的二进制编码 00，01，10，11 分别示A，B，C，D. 称这种表示法为等长码表示法.

若在传输中，A，B，C，D出现的频率大体相同，用等长码表示是很好的方法. 但当它们出现的频率相差悬殊时，为了节省二进制数位，以达到提高效率的目的，就要采用非等长的编码.

二元前缀码

定义16.10 设$\alpha_1\alpha_2\ldots\alpha_{n-1}\alpha_n$是长为 n 的符号串

(1) 前缀——$\alpha_1, \alpha_1\alpha_2, \ldots, \alpha_1\alpha_2\ldots\alpha_{n-1}$

(2) 前缀码A——$A=\{\beta_1, \beta_2, \ldots, \beta_m\}$中任何两个元素互不为前缀

(3) 二元前缀码——由0-1符号串构成的前缀码.

例 {1，00，011，0101，01001，01000}为前缀码；

而{1，00，011，0101，0100，01001，01000}不是前缀码，因为0100既是01001又是01000的前缀。

利用2 叉树产生2元前缀码

如何产生2元前缀码？可以用2 叉树产生2元前缀码.

设T是具有n 片树叶的2 叉树，则T的每个分支点有1或2个儿子. 设v为T的分支点，若v有两个儿子，在由v引出的两条边上，左边的标0，右边的标1. 若只有一个儿子，由它引出的边可以标0，也可以标1.

设v_i是T的任意一片树叶，从树根到v_i的通路上各边的标号(0或1)按照通路上边的顺序组成的符号串放在v_i处，t片树叶的t个符号串组成一个2元前缀码.

利用2 叉树产生2元前缀码

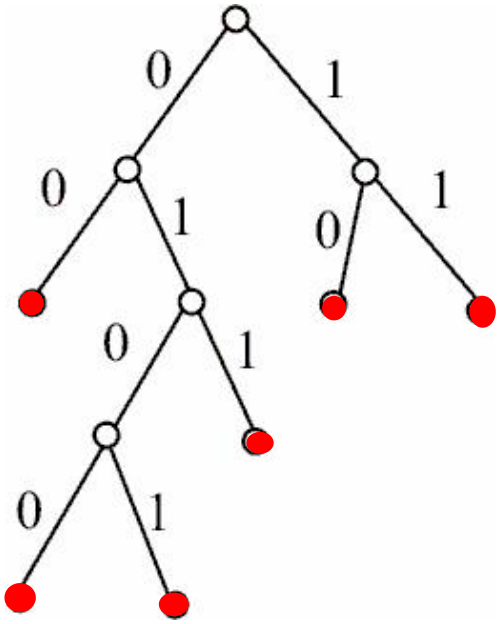
定理16.6 一棵2叉树可以产生一个二元前缀码，但不一定唯一.

推论 一棵2叉正则树可以产生惟一的前缀码.

利用2 叉树产生2元前缀码

图所示2叉正则树产生的前缀码为：

{ 00, 10, 11, 011, 0100, 0101 }



可以用上述前缀码来传输6个符号，如 A~F. 但当在文本中这些字母出现的频率不同时，传输这个文本所用的二进制位数是不同的.

显然，频率越高的字母应用二进制位数越少的符号串传输，整个文本的传输耗存才会最小.

最佳前缀码

最佳前缀码 —— 以各个符号出现的频率为权的最优 2 叉树产生的前缀码

例8 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？

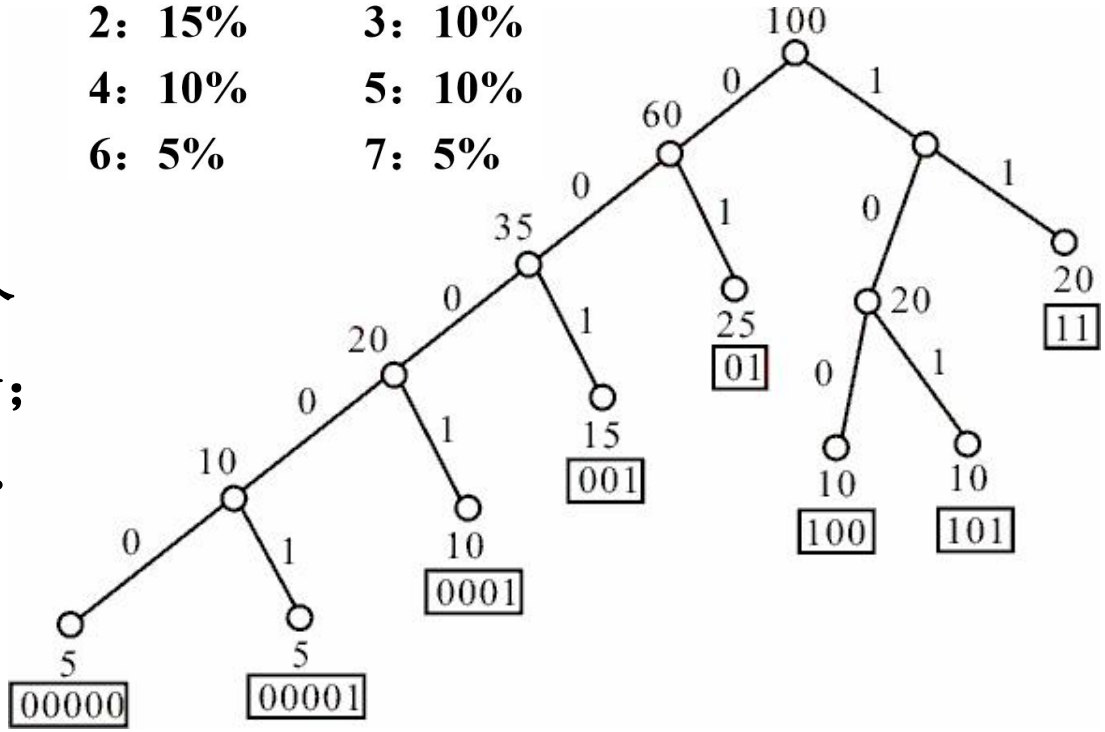
求最佳前缀码

解 用100个八进制数字中各数字出现的个数，即以100乘各频率为权，并将各权由小到大排列，得$w_1=5$, $w_2=5$, $w_3=10$, $w_4=10$, $w_5=10$, $w_6=15$, $w_7=20$, $w_8=25$. 用此权产生的最优树如图所示.

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

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

$W(T)=285$，传$10^n(n\geq 2)$个用二进制数字2.85×10^n个；用等长码需3×10^n个数字.



波兰符号法与逆波兰符号法

行遍或周游根树T——对T的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——次序为：左子树、根、右子树

② 前序行遍法——次序为：根、左子树、右子树

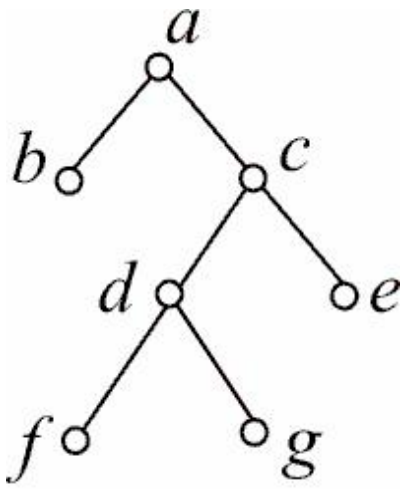
③ 后序行遍法——次序为：左子树、右子树、根

对图所示根树按中序、前序、后序行遍法访问结果分别为：

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

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

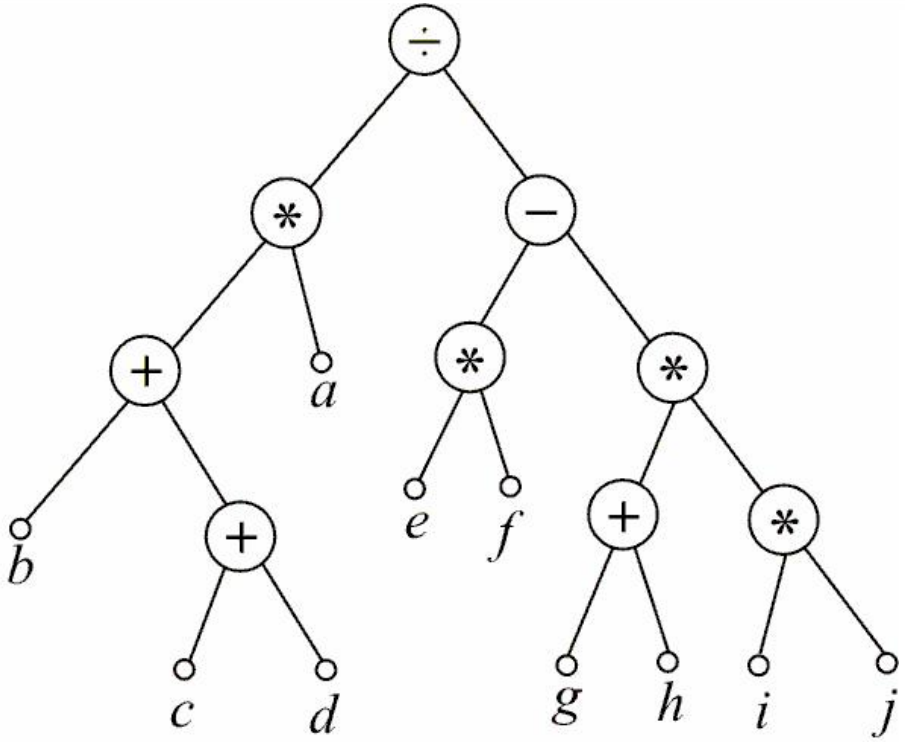
$b\ ((f\ g\ \underline{d})\ e\ \underline{c})\ \underline{a}$ (后)



用2叉有序正则树存放算式

存放规则

- 最高层次运算放在树根
- 后依次将运算符放在根子树的根上
- 数放在树叶上
- 规定：被除数、被减数放在左子树树叶上



用3种行遍法访问这颗2叉树：

(1)中序行遍法：还原算式

算式 $((b+(c+d))*a)\div((e*f)-(g+h)*(i*j))$存放在图所示2叉树上.

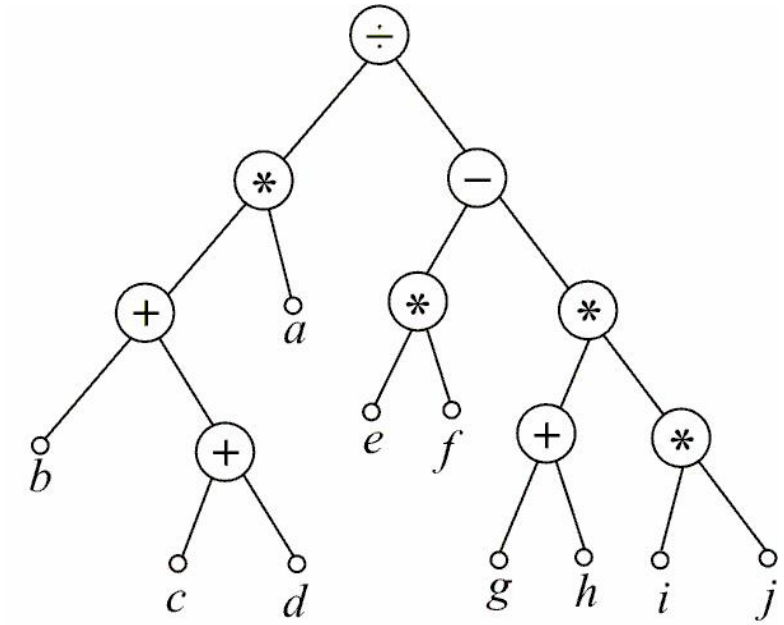
波兰符号法

波兰符号法 (前缀符号法)

(2)前序行遍法：访问存放算式的2叉有序正则树，其结果不加括号，规定从右到左每个运算符号对它后面紧邻的两个数进行运算，运算结果正确.

称此算法为波兰符号法或前缀符号法. 对上图的访问结果为

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



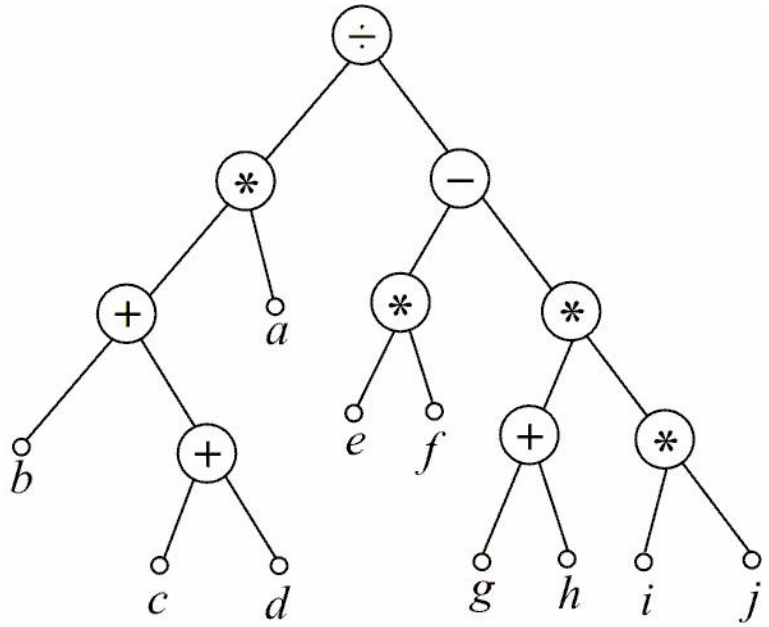
逆波兰符号法

逆波兰符号法 (后缀符号法)

(3)后序行遍法：规定从左到右每个运算符与前面紧邻两数运算.

称此算法为逆波兰符号法或后缀符号法. 对上图的访问结果为

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



最佳前缀码练习题

练习 设7个字母在通信中出现的频率如下：

A: 35%，B:20%，C:15%，D:10%，E:5%，F:5%，G:10%

利用Huffman算法求传输它们的最佳二进制前缀码。

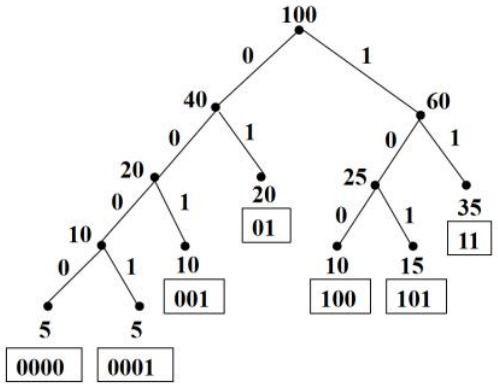
要求： （1）画出最优树，指出每个字母对应的编码，

（2）计算传输10^n（$n\geq 2$）个按上述频率出现的字母需要多少个二进制数字。

练习答案

解：

(1) 最优二叉树：（画法不唯一，但权重唯一）



（3 分）

(2) 每个字母对应的编码：（与最优二叉树对应即可）

A: 11　B: 01　C: 101　D: 100　E: 0000　F: 0001　G: 001　（6 分）

(3) 需要多少个二进制数字：

W(T) = 10+20+40+25+60+100 = 255，　故传输 10^n ($n \geq 2$)个按上述频率出现的字母需要 2.55×10^n。　（8 分）