



二叉树





定义9.9 设二叉树 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

$$W(t) = \sum_{i=1}^t w_i l(v_i)$$

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, \dots, w_t 的二叉树中权最小的二叉树称为**最优二叉树**.





求最优树的算法—— Huffman算法

给定实数 w_1, w_2, \dots, w_t , 且 $w_1 \leq w_2 \leq \dots \leq w_t$.

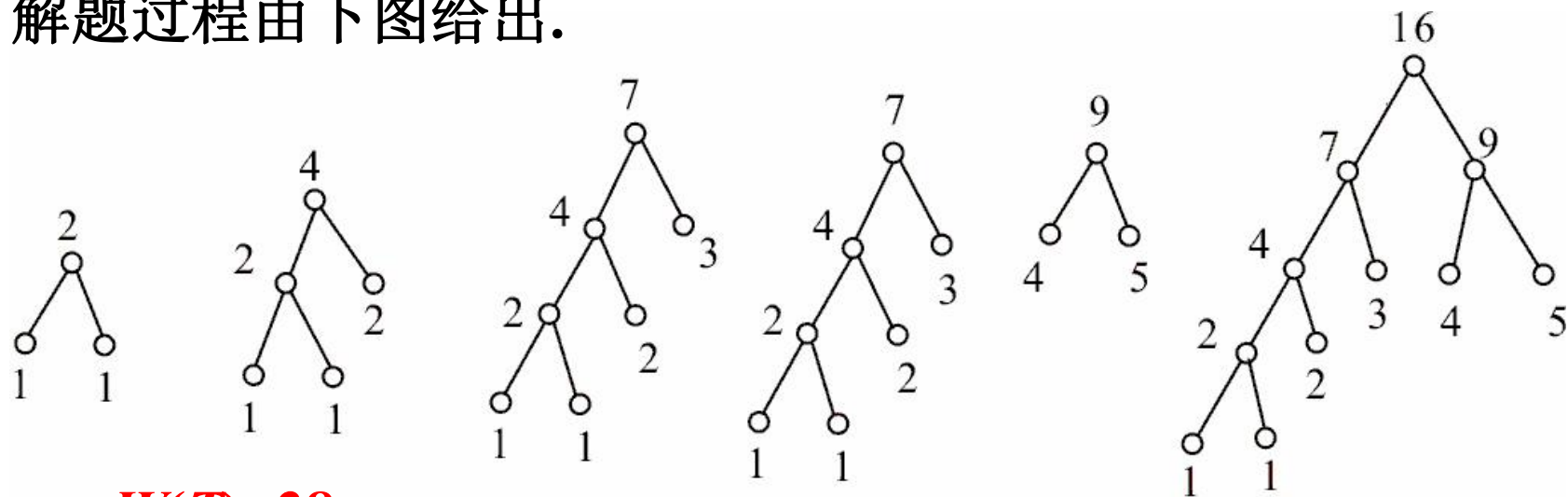
- (1) 连接权为 w_1, w_2 的两片树叶, 得一个分支点, 其权为 $w_1 + w_2$.
- (2) 在 $w_1 + w_2, w_3, \dots, w_t$ 中选出两个最小的权, 连接它们对应的顶点(不一定是树叶), 得新分支点及所带的权.
- (3) 重复(2), 直到形成 $t-1$ 个分支点, t 片树叶为止.





例 5 求带权为1, 1, 2, 3, 4, 5的最优树.

解题过程由下图给出.



$$W(T)=38$$





通信中的非等长编码问题

定义9.10 设 $\alpha_1 \alpha_2 \dots \alpha_{n-1} \alpha_n$ 是长度为 n 的符号串

- (1) **前缀**——以上符号串的子串 $\alpha_1, \alpha_1 \alpha_2, \dots, \alpha_1 \alpha_2 \dots \alpha_n$
- (2) **前缀码**—— $\{\beta_1, \beta_2, \dots, \beta_m\}$ 为符号串的集合,若其中任何两个符号串互不为前缀
- (3) **二元前缀码**—— $\beta_i (i=1, 2, \dots, m)$ 中只出现两个符号, 如0与1.

如: $\{1, 00, 011, 0101, 01001, 01000\}$ 为前缀码

$\{1, 00, 011, 0101, 0100, 01001, 01000\}$ 呢?

如何产生二元前缀码?





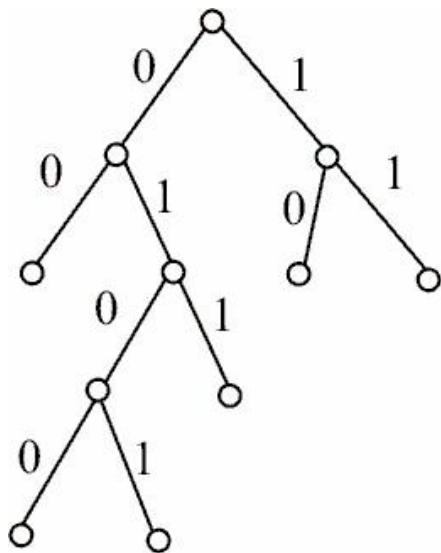
定理9.5 在二叉树中按左子树标0，右子树标1；若只有一个子树，则可以标0，也可以标1。这样，从树根到叶子的通路上所有标号按通路上的顺序组成的符号串的集合，构成二元前缀码（包含叶子数个码串）。

推论 一棵正则2叉树产生唯一的前缀码。





图所示二叉树产生的前缀码为
 $\{ 00, 10, 11, 011, 0100, 0101 \}$



前缀码的数量与对应二叉树的叶子树相等





- (1) 前缀码可以代表通信中传输的符号
- (2) 当需要传输的文本仅由这些符号构成，则每个符号出现的频率不同，希望传输这个文本时代表每个符号的二进制位数按其出现频率的不同而不同
- (3) 设共有 t 个符号，树叶 v_i 对应的前缀码二进制串表示的符号的出现频率为 c_i ， v_i 对应的前缀码的长度等于 v_i 的层数 $l(v_i)$ ，因此传输 m 个符号使用的二进制位数为 $m \sum c_i l(v_i)$ 一个符号的平均长度
- (4) 用以各符号出现的频率为权的最优二叉树产生的前缀码所用的二进制位数最少！

称这个由最优二叉树产生的前缀码为最佳前缀码

用最佳前缀码
传输的二进制
位数最省！！





例6 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？



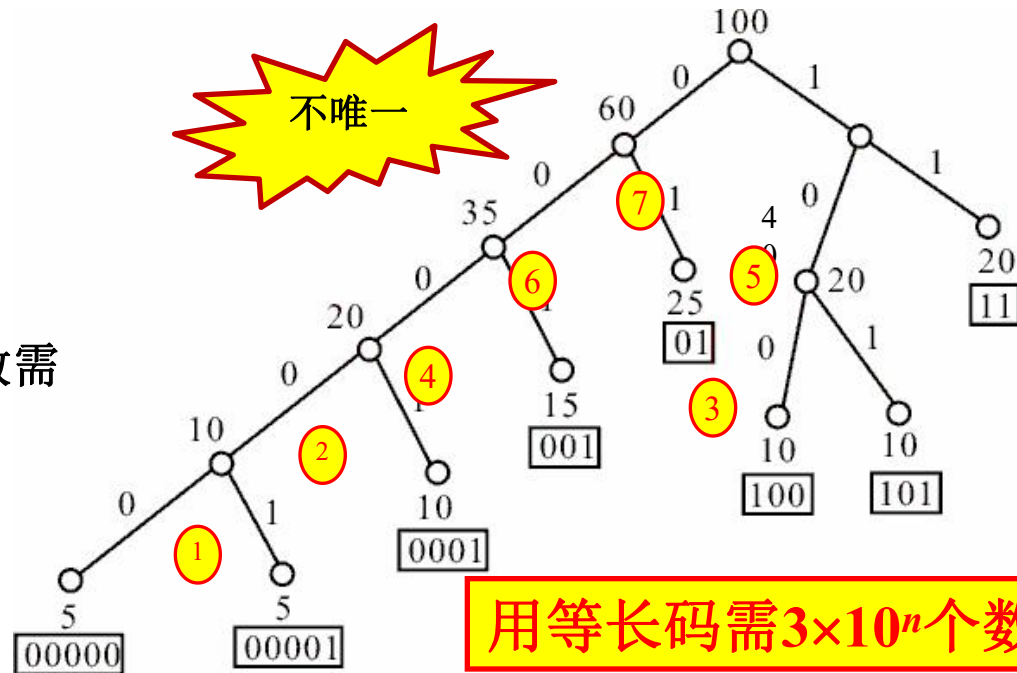


解 用100个八进制数字中各数字出现的个数，即以100乘各频率为权，并将各权由小到大排列，得 $w_1=5, w_2=5, w_3=10, w_4=10, w_5=10, w_6=15, w_7=20, w_8=25$. 用此权产生的最优2叉树如图所示.

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

$W(T)=285$ (100个八进制数需要285个二进制数来表示)

传 $10^n (n \geq 2)$ 个
用二进制数字需
 2.85×10^n 个.





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：

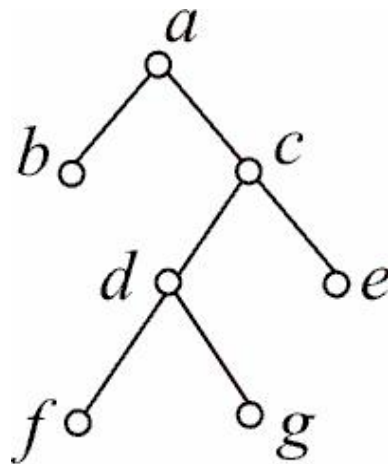
- ① 中序行遍法——次序为：左子树、根、右子树
- ② 前序行遍法——次序为：根、左子树、右子树
- ③ 后序行遍法——次序为：左子树、右子树、根

对图所示根树按中序、前序、后序行遍法访问结果分别为：

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

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

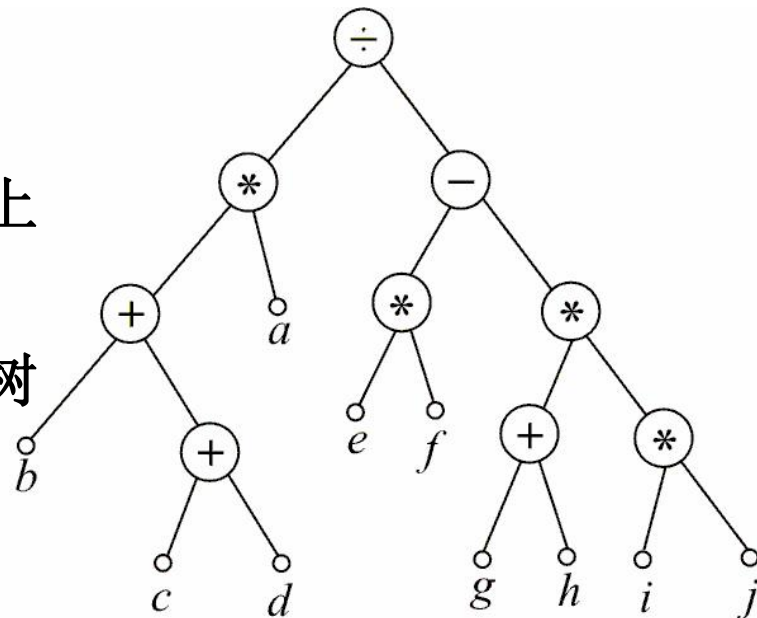
$b ((f g \underline{d}) e \underline{c}) \underline{a}$





存放规则

- 最高层次运算放在树根
- 然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



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





THE END

