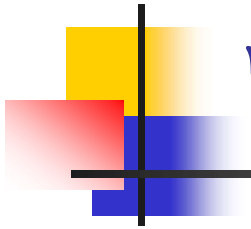
哈夫曼编码

- 文件的压缩存储

1. 定长编码：3位0-1字符串
2. 不定长编码：压缩约25%

	a	b	c	d	e	f
Frequency (in thousands)	45	13	12	16	9	5
Fixed-length codeword	000	001	010	011	100	101
Variable-length codeword	0	101	100	111	1101	1100

Figure 16.3 A character-coding problem. A data file of 100,000 characters contains only the characters a–f, with the frequencies indicated. If we assign each character a 3-bit codeword, we can encode the file in 300,000 bits. Using the variable-length code shown, we can encode the file in only 224,000 bits.



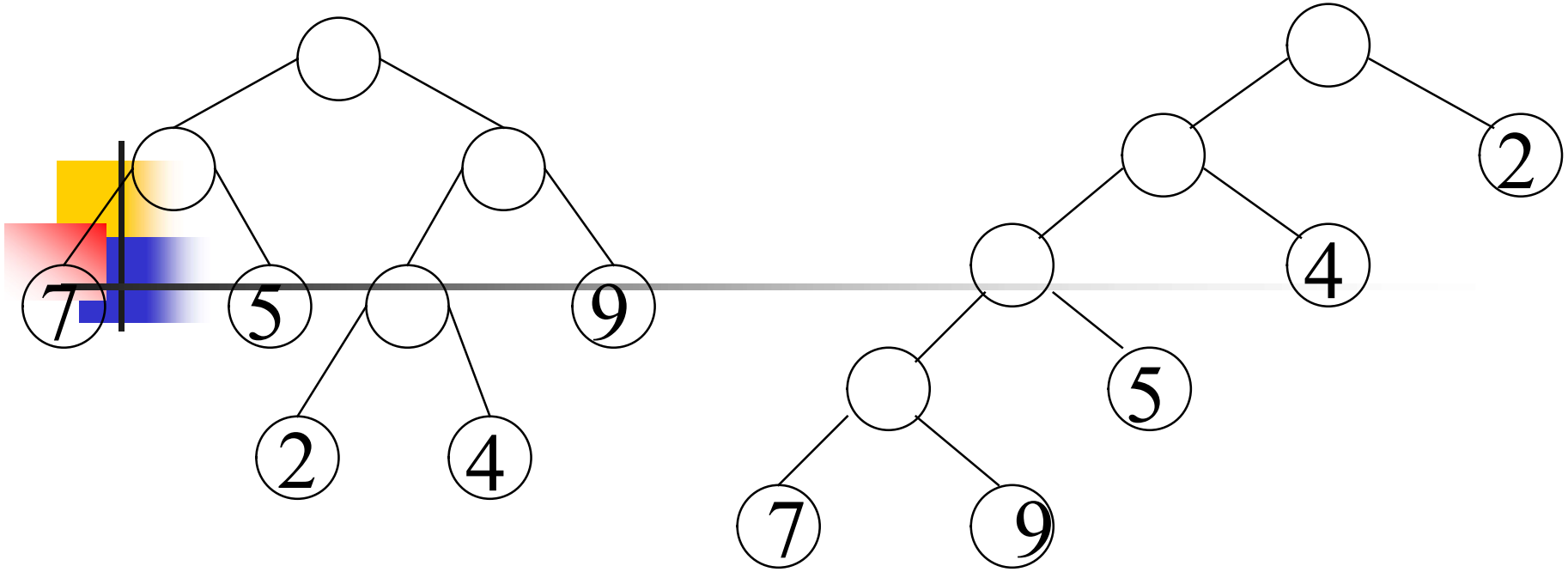
哈夫曼编码

- 前缀编码：任何一个字符的编码都不是同一字符集中另一个字符的编码的前缀。
- 二叉树构造前缀编码：interpret the binary codeword for a character as the simple path from the root to that character, where 0 means “go to the left child” and 1 means “go to the right child” .

哈夫曼编码

- **如何构造最优二叉树----赫夫曼算法**
  - **输入：** n 和 n 个权值 $\{w_1, w_2, \ldots, w_n\}$
  - **输出：** 赫夫曼树

1. 根据给定的 n 个权值 $\{w_1, w_2, \ldots, w_n\}$，构造 n 棵二叉树的集合 $F = \{T_1, T_2, \ldots, T_n\}$，其中每棵二叉树中均只含一个带权值为 w_i 的根结点，其左、右子树为空树；
2. 在 F 中选取其根结点的权值为**最小**和**次小**的两棵二叉树，分别作为左、右子树构造一棵新的二叉树，并置这棵新的二叉树根结点的权值为其左、右子树根结点的权值之和；
3. 从 F 中**删去**这两棵树，同时**加入**刚生成的新树；
4. 重复 2 和 3 两步，**直至 F 中只含一棵树为止**



B(T)=

7×2+5×2+2×3+

4×3+9×2

=60

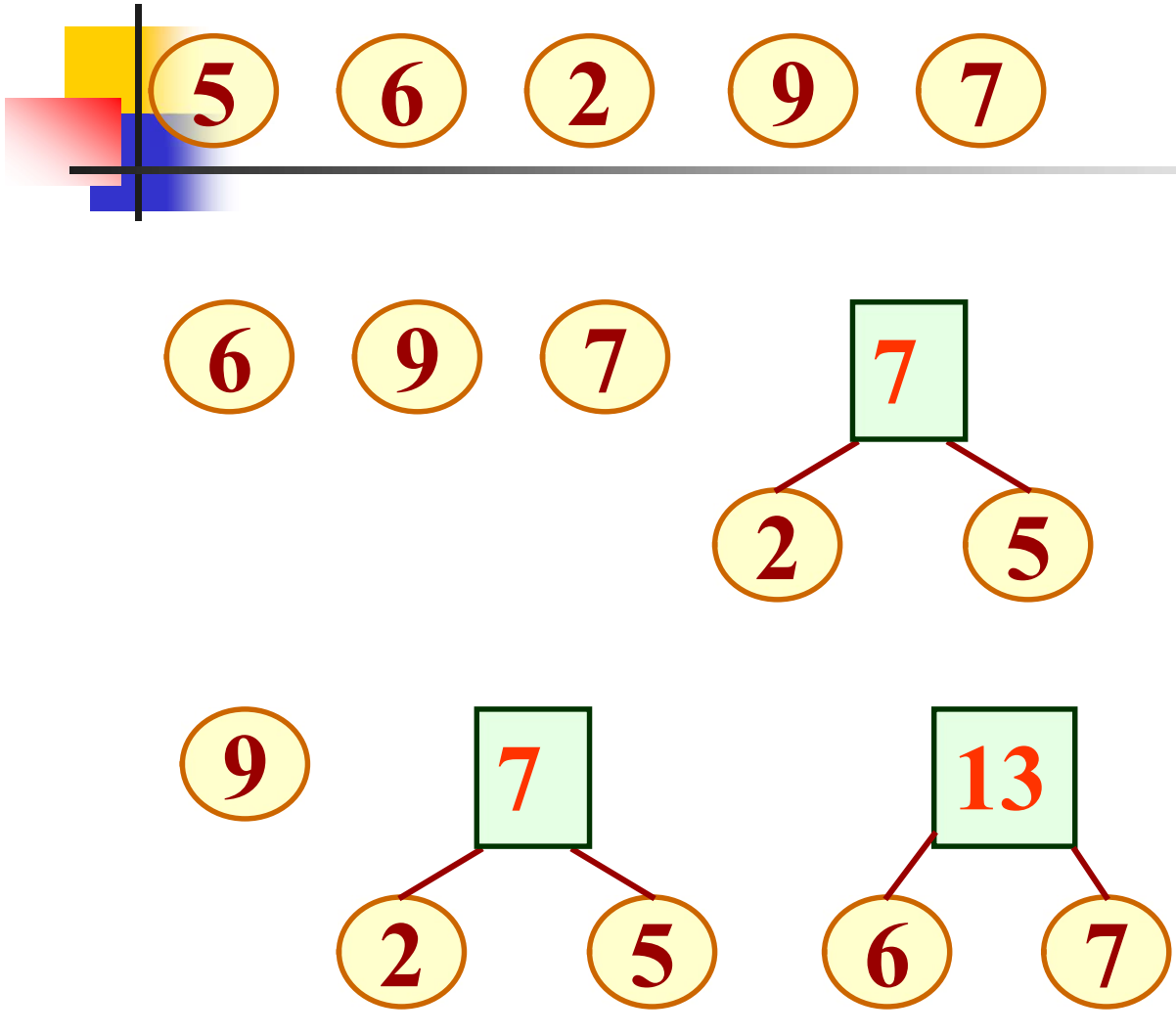
B(T)=

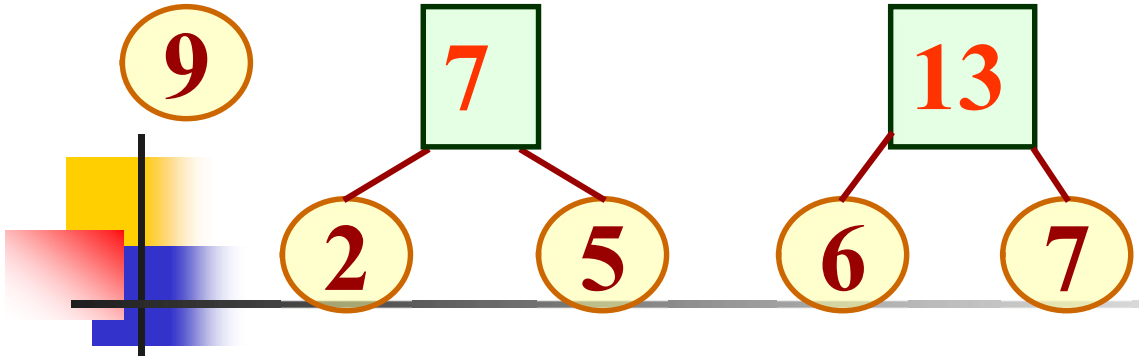
7×4+9×4+5×3+

4×2+2×1

=89

例如：已知权值 W={5, 6, 2, 9, 7}

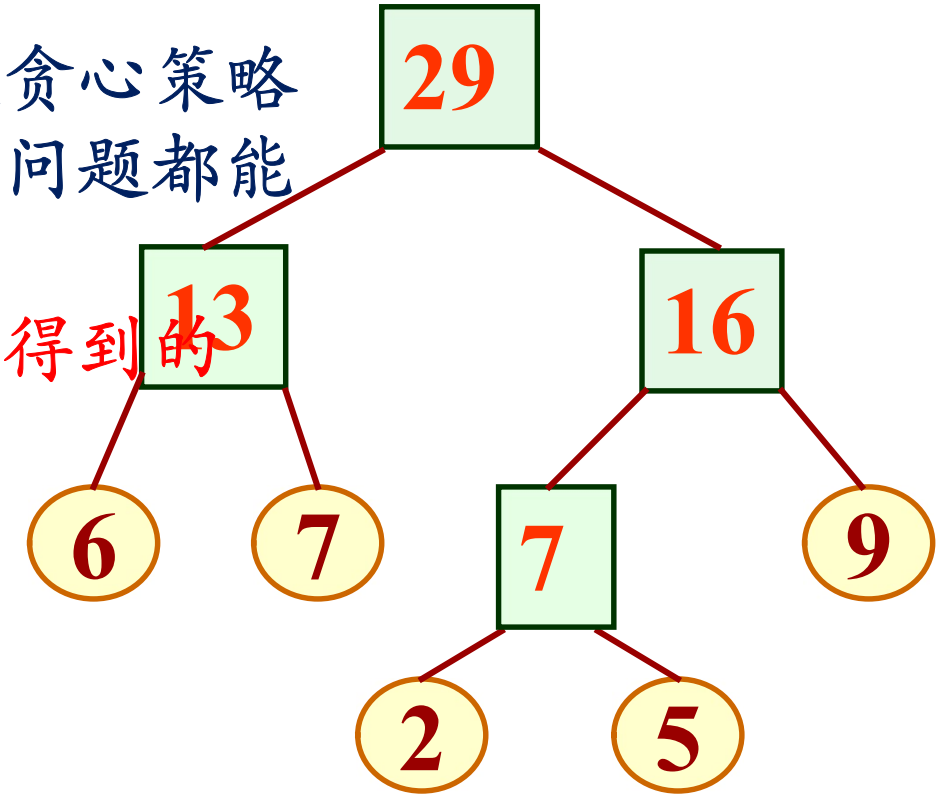




赫夫曼算法采用的是贪心策略

贪心策略不是对所有问题都能得到最优解

可以证明赫夫曼算法得到的是最优解



哈夫曼算法—正确性证明

- 引理8.8:

  Let C be an alphabet in which each character $c \in C$ has frequency $c.freq$. Let x and y be two characters in C having the lowest frequencies. Then there exists an optimal prefix code for C in which the codewords for x and y have the same length and differ only in the last bit.

  - 令C为字符集，任一字符c∈C的频率为c.freq，x和y是C中频率最小的2个字符。那么存在一个C的最优前缀编码使得x和y的编码长度相同，并且只有最后一位不同。

- 用二叉树构造编码----存在度为1的结点吗?

哈夫曼算法—正确性证明

分析：

- The idea of the proof is to take the tree T representing an arbitrary optimal prefix code and modify it to make a tree representing another optimal prefix code such that the characters x and y appear as sibling leaves of maximum depth in the new tree. If we can construct such a tree, then the codewords for x and y will have the same length and differ only in the last bit.

1. 存在最优二叉树， x 和 y是路径长度最长的叶结点，且是兄弟
2. 任一棵最优二叉树都可经过适当的“改造”，构造这样一棵最优二叉树

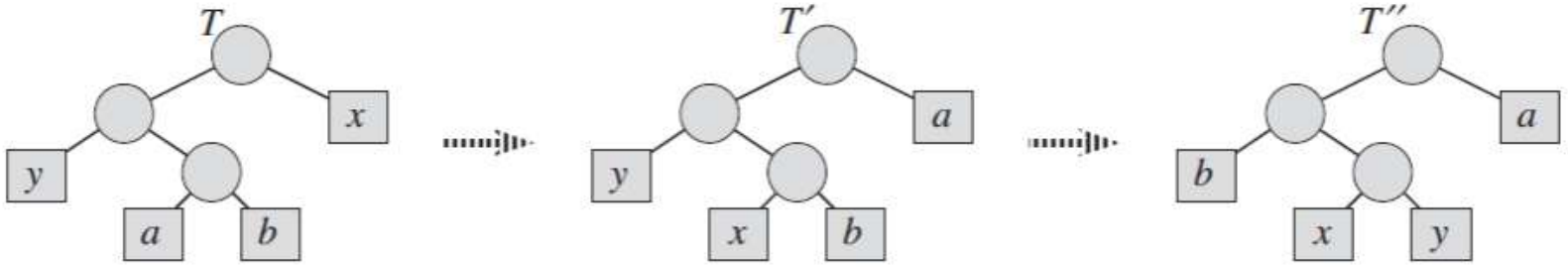
哈夫曼算法—正确性证明

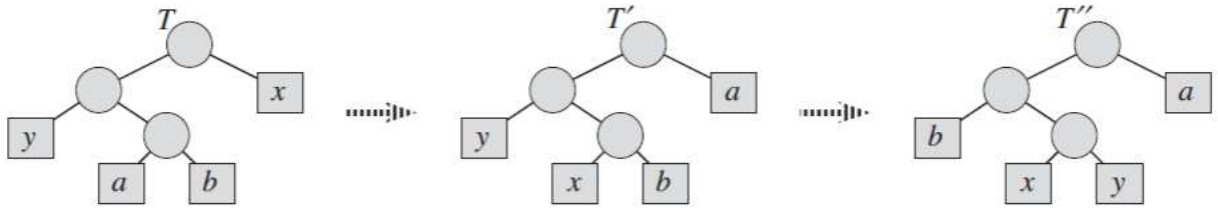
- 证明：令T是字符集C的一棵最优二叉树。x和y是C中频率最小的2个字符，不失一般性我们令$x.freq \le y.freq$.

1. **Let *a* and *b* be two characters that are sibling leaves of maximum depth in T. Without loss of generality, we assume that $a.freq \le b.freq$.**

哈夫曼算法—正确性证明

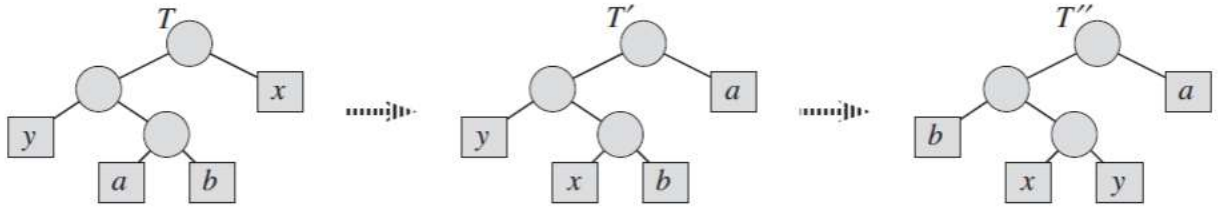
- **Since *x* and *y* have the lowest frequencies, *a.freq* and *b.freq* are two arbitrary frequencies, we have that:**
- ***x.freq* ≤ *a.freq*, *y.freq* ≤ *b.freq*.**
- 交换T中a和x的位置，得到一棵新树 T'；交换T'中b和y的位置，得到一棵新树T''。





- **in T'', x and y are sibling leaves of maximum depth.**
- **the difference in cost between T and T':**
- $B(T) - B(T')$

$$
\begin{aligned}
&= \sum_{c \in C} c.freq \cdot d_T(c) - \sum_{c \in C} c.freq \cdot d_{T'}(c) \\
&= x.freq \cdot d_T(x) + a.freq \cdot d_T(a) - x.freq \cdot d_{T'}(x) - a.freq \cdot d_{T'}(a) \\
&= x.freq \cdot d_T(x) + a.freq \cdot d_T(a) - x.freq \cdot d_T(a) - a.freq \cdot d_T(x) \\
&= (a.freq - x.freq)(d_T(a) - d_T(x)) \\
&\geq 0 ,
\end{aligned}
$$



$$
\begin{aligned}
& B(T') - B(T'') \\
&= \sum_{c \in C} c.freq \times d_{T'}(c) - \sum_{c \in C} c.freq \times d_{T''}(c) \\
&= y.freq \times d_{T'}(y) + b.freq \times d_{T'}(b) - y.freq \times d_{T''}(y) - b.freq \times d_{T''}(b) \\
&= y.freq \times d_{T'}(y) + b.freq \times d_{T'}(b) - y.freq \times d_{T'}(b) - b.freq \times d_{T'}(y) \\
&= b.freq(d_{T'}(b) - d_{T'}(y)) - y.freq(d_{T'}(b) - d_{T'}(y)) \\
&= (b.freq - y.freq)(d_{T'}(b) - d_{T'}(y)) \\
&\geq 0
\end{aligned}
$$

- 交换b和y的位置，没有增加树的代价。

哈夫曼算法—正确性证明

- Therefore, $B(T) \geq B(T'')$ and since T is optimal, we have $B(T) \leq B(T'')$, which implies $B(T) = B(T'')$ .Thus, $\boldsymbol{T''}$ is an optimal tree in which x and y appear as sibling leaves of maximum depth, from which the lemma follows.

Let C be an alphabet in which each character $c \in C$ has frequency $c.freq$. Let x and y be two characters in C having the lowest frequencies. Then there exists an optimal prefix code for C in which the codewords for x and y have the same length and differ only in the last bit.

哈夫曼算法—正确性证明

- 定理8.9：Let C be a given alphabet with frequency $c.freq$ defined for each character $c \in C$.
  - Let x and y be two characters in C with minimum frequency. Let C' be the alphabet $C' = C - \{x, y\} + \{z\}$, $z.freq = x.freq + y.freq$.
  - Let T' be any tree representing an optimal prefix code for the alphabet C'. Then the tree T, obtained from T' by replacing the leaf node for z with an internal node having x and y as children, represents an optimal prefix code for the alphabet C.

哈夫曼算法—正确性证明

- **We first show how to express the cost $B(T)$ of tree T in terms of the cost $B(T')$ of tree T'**

1. **for each $c \in C-\{x,y\}$, $d_T(c)=d_{T'}(c)$, $c.freq \times d_T(c)=c.freq \times d_{T'}(c)$**
2. **$d_T(x)=d_T(y)=d_{T'}(z)+1$**
3. **$x.freq \times d_T(x)+y.freq \times d_T(y)=(x.freq+y.freq)(d_{T'}(z)+1)$**
   $=z.freq \times d_{T'}(z)+x.freq+y.freq$

$$B(T)=\sum_{c\in C}(c.freq \times d_T(c))$$

$$B(T)=\sum_{c\in C-\{x,y\}}(c.freq \times d_T(c))+x.freq \times d_T(x)+y.freq \times d_T(y)$$

$$B(T)=\sum_{c\in C-\{x,y\}}(c.freq \times d_T(c))+z.freq \times d_{T'}(z)+x.freq+y.freq$$

$$B(T)=B(T')+x.freq+y.freq$$

哈夫曼算法—正确性证明

$$B(T) = B(T') + x.freq + y.freq$$

- 采用反证法。假设T不代表C的最优前缀编码，那么存在C的最优编码树T"，则B(T)>B(T")，且x和y是兄弟结点
- 令T""为由树T"中将x和y的双亲结点替换成叶结点z, z.freq=x.freq+y.freq

$$\begin{aligned} B(T''') &= B(T'') - x.freq - y.freq \\ &< B(T) - x.freq - y.freq \\ &= B(T'), \end{aligned}$$

yielding a contradiction to the assumption that T' represents an optimal prefix code for C'. Thus, T must represent an optimal prefix code for the alphabet C. ■

哈夫曼算法—正确性证明

- 对字符集中的字符数n采用数学归纳法
  - n=2时，两个字符都是1位编码，哈夫曼树构造的就是最优的编码
  - 令n<k时哈夫曼算法正确。那么当n=k时，字符集C中有k个字符，频率最低的2个字符不妨设为x和y。

1. 哈夫曼算法首先对每个字符构造一个只有一个根结点的树，每棵树根结点的权值就是对应字符的频率，k棵树构成森林Forest。
2. 根据哈夫曼算法，选择根结点权值最小的两棵树分别作为左右子树构造一棵新的二叉树，新树根结点的权值为其左右子树的权值之和。即：将x和y对应的树分别作为左右子树生成一个新树z，且z.freq=x.freq+y.freq. 则Forest中有k-1棵树，除了树z，每棵树对应C中的一个字符，只有一个根结点。.....