

本节内容

# 哈夫曼树

# 知识总览

## 哈夫曼树

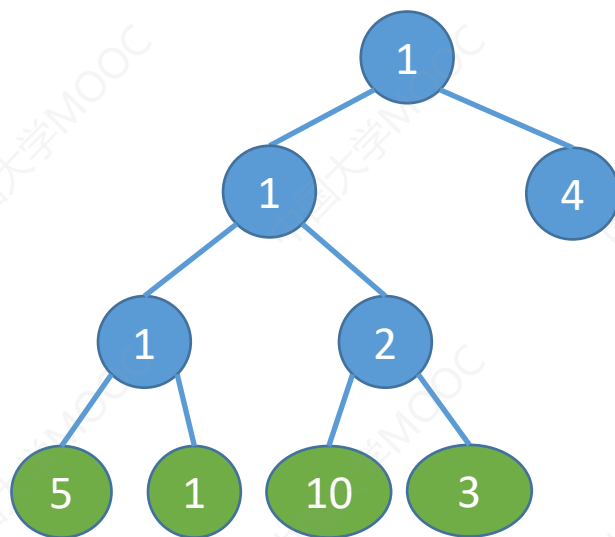
带权路径长度

哈夫曼树的定义

哈夫曼树的构造

哈夫曼编码

# 带权路径长度



结点的**权**：有某种现实含义的数值（如：表示结点的重要性等）

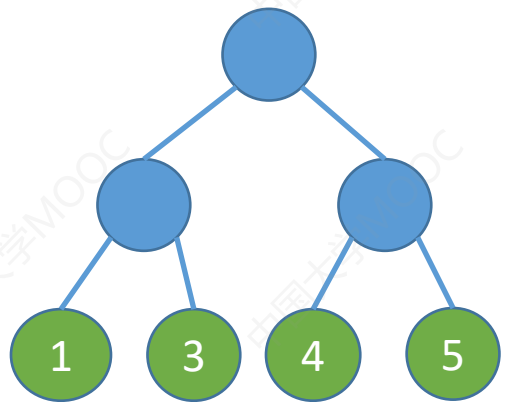
**结点的带权路径长度**：从树的根到该结点的路径长度（经过的边数）与该结点上权值的乘积

**树的带权路径长度**：树中所有**叶结点**的带权路径长度之和（WPL, Weighted Path Length）

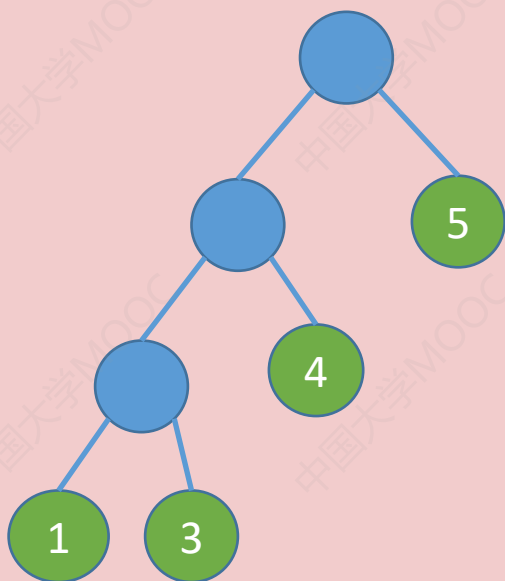
$$WPL = \sum_{i=1}^n w_i l_i$$

## 哈夫曼树的定义

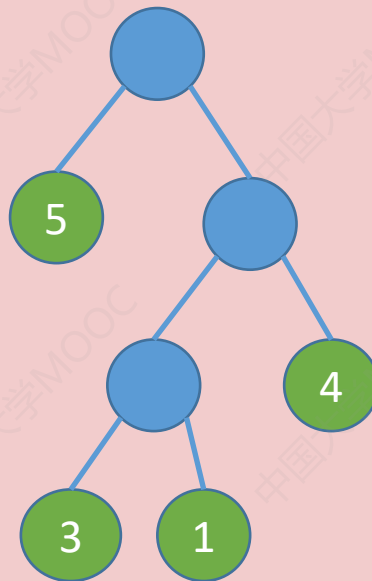
都是哈夫曼树



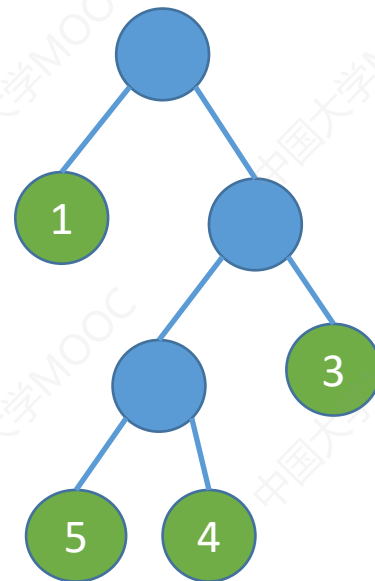
$$WPL = 2 \times 1 + 2 \times 3 + 2 \times 4 + 2 \times 5 = 26$$



$$WPL = 1 \times 5 + 2 \times 4 + 3 \times 1 + 3 \times 3 = 25$$



$$WPL = 1 \times 5 + 2 \times 4 + 3 \times 1 + 3 \times 3 = 25$$



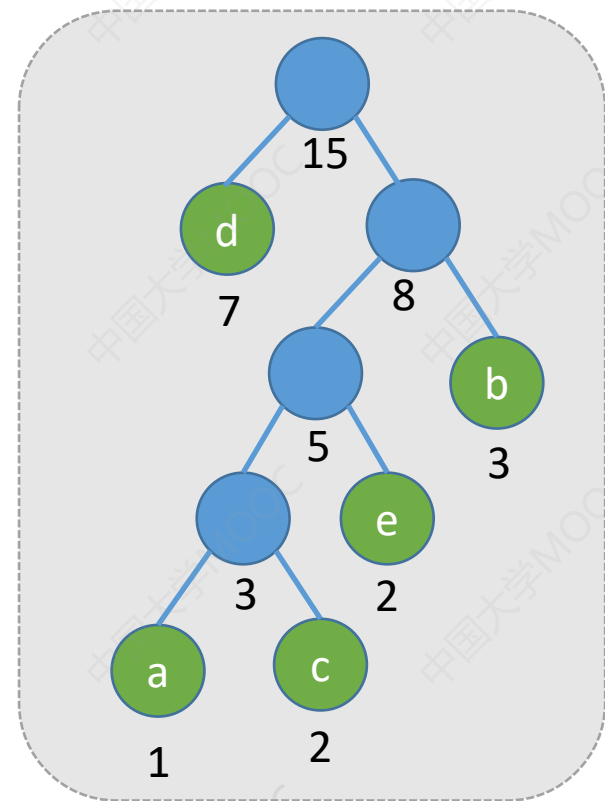
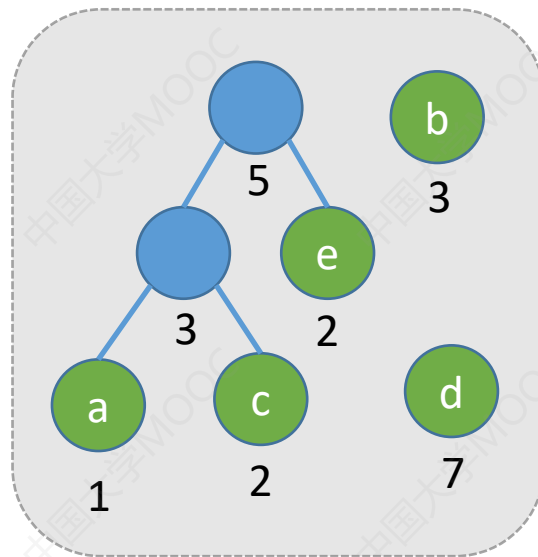
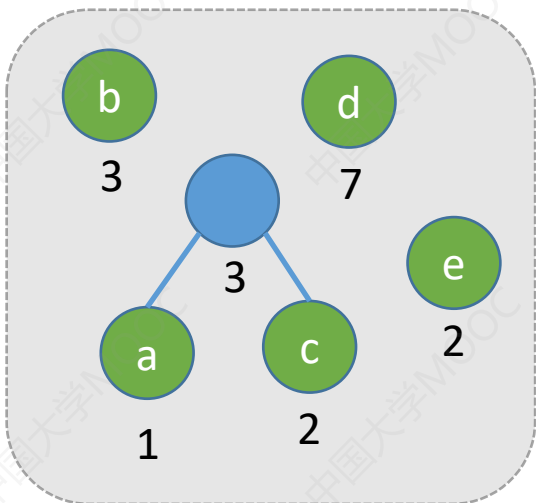
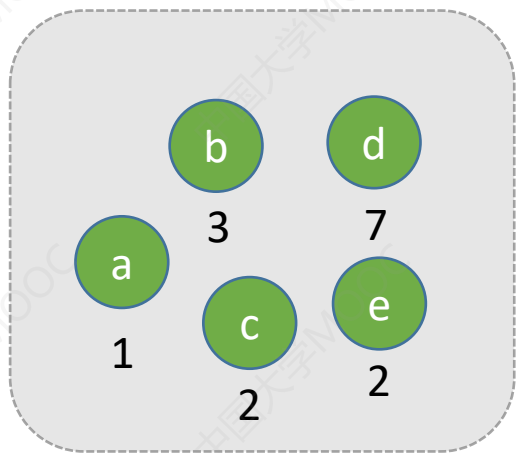
$$WPL = 1 \times 1 + 2 \times 3 + 3 \times 5 + 3 \times 4 = 34$$

在含有 $n$ 个带权叶结点的二叉树中，其中带权路径长度（WPL）最小的二叉树称为哈夫曼树，也称最优二叉树

## 哈夫曼树的构造

给定 $n$ 个权值分别为 $w_1, w_2, \dots, w_n$ 的结点，构造哈夫曼树的算法描述如下：

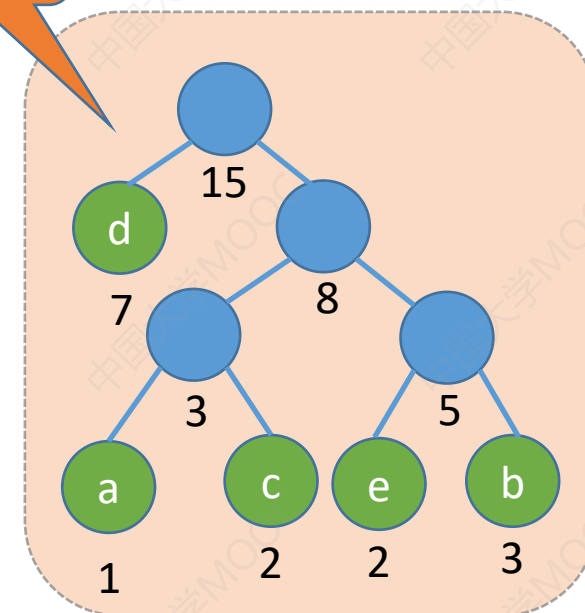
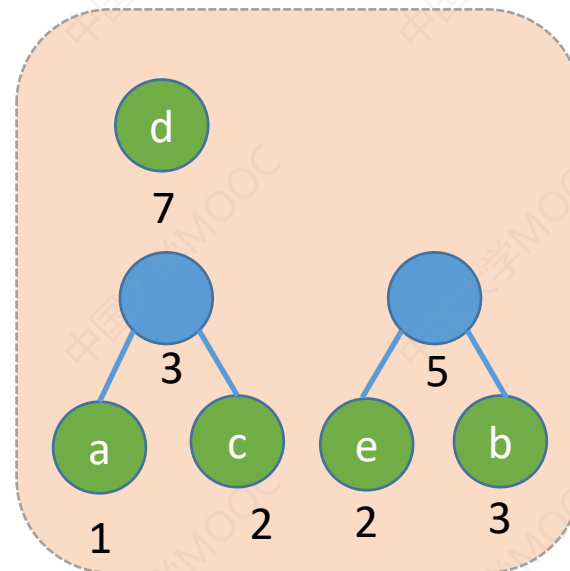
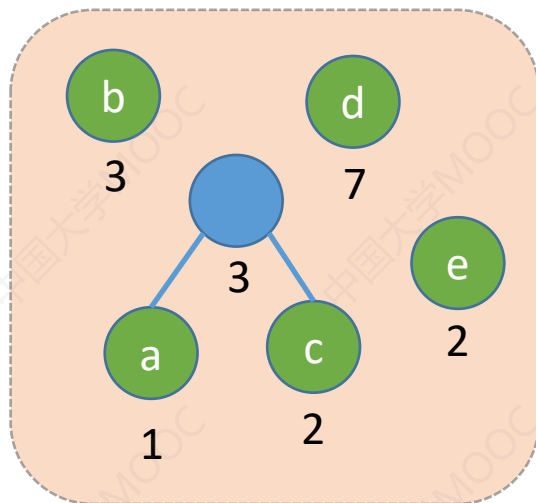
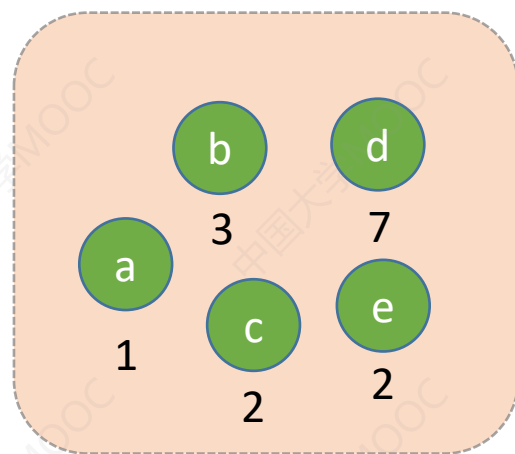
- 1) 将这 $n$ 个结点分别作为 $n$ 棵仅含一个结点的二叉树，构成森林 $F$ 。
- 2) 构造一个新结点，从 $F$ 中选取两棵根结点权值最小的树作为新结点的左、右子树，并且将新结点的权值置为左、右子树上根结点的权值之和。
- 3) 从 $F$ 中删除刚才选出的两棵树，同时将新得到的树加入 $F$ 中。
- 4) 重复步骤2) 和3)，直至 $F$ 中只剩下一棵树为止。



- 1) 每个初始结点最终都成为叶结点，且权值越小的结点到根结点的路径长度越大
- 2) 哈夫曼树的结点总数为 $2n - 1$
- 3) 哈夫曼树中不存在度为1的结点。
- 4) 哈夫曼树并不唯一，但WPL必然相同且为最优

$$WPL_{\min} = 1 \times 7 + 2 \times 3 + 3 \times 2 + 4 \times 1 + 4 \times 2 = 31$$

# 哈夫曼树的构造



$$WPL = 1 \times 7 + 3 \times (1 + 2 + 2 + 3) = 31$$

## 哈夫曼编码



电报——点、划 两个信号（二进制0/1）

# 哈夫曼编码

固定长度编码——每个字符用相等长度的二进制位表示

A——0100 0001  
B——0100 0010  
C——0100 0011  
D——0100 0100

ASCII编码

A——00  
B——01  
C——10  
D——11

每个字符用长度为2的二进制表示

假设，100题中有80题选C，10题选A，8题选B，2题选D

所有答案的二进制长度=80\*2+10\*2+8\*2+2\*2=200 bit

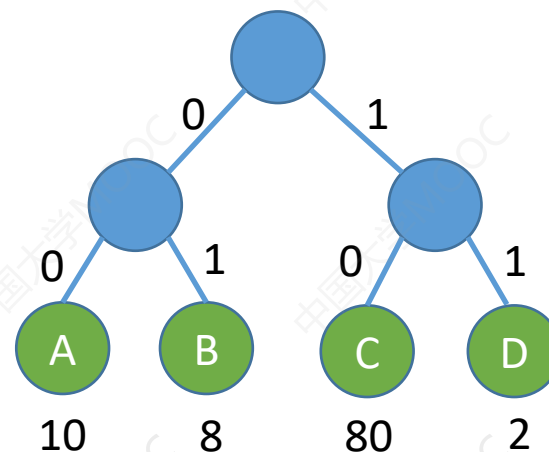


小渣

100个选择题



老渣



$$WPL = 80 \times 2 + 10 \times 2 + 8 \times 2 + 2 \times 2 = 200$$

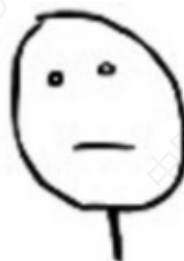


# 哈夫曼编码

固定长度编码——每个字符用相等长度的二进制位表示

A—00  
B—01  
C—10  
D—11

每个字符用长度为2的二进制表示



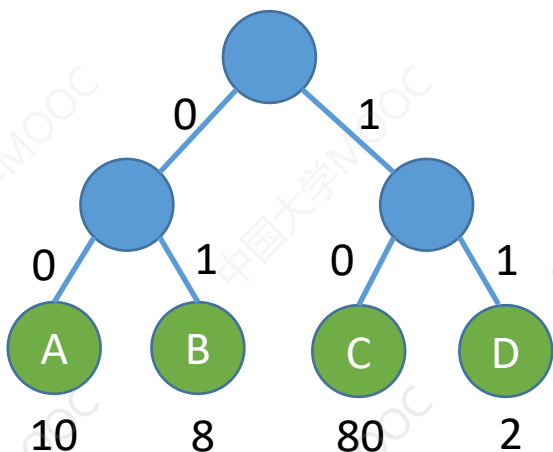
小渣

100个选择题



老渣

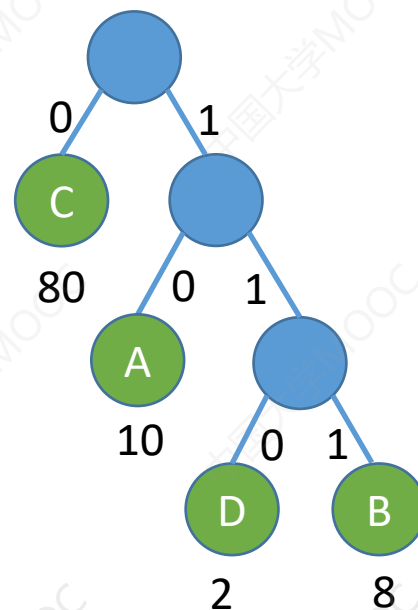
假设，100题中有80题选C，10题选A，8题选B，2题选D  
所有答案的二进制长度=80\*2+10\*2+8\*2+2\*2=200 bit



$$WPL = 80 \times 2 + 10 \times 2 + 8 \times 2 + 2 \times 2 = 200$$

C—0  
A—10  
B—111  
D—110

可变长度编码——允许对不同字符用不等长的二进制位表示



$$WPL = 80 \times 1 + 10 \times 2 + 2 \times 3 + 8 \times 3 = 130$$

# 哈夫曼编码

可变长度编码——允许对不同字符用不等长的二进制位表示

若没有一个编码是另一个编码的前缀，则称这样的编码为前缀编码

好的收到，CBBD



小渣

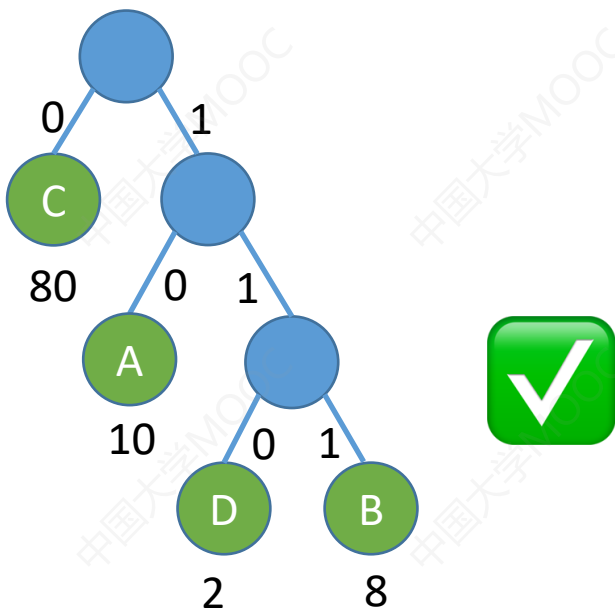
CAAABD



老渣

前缀码解  
码无歧义

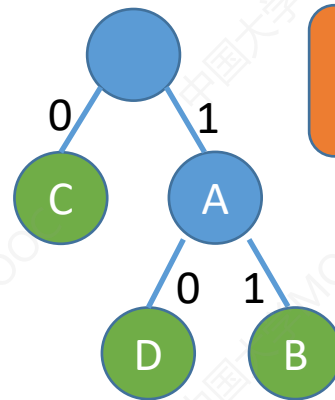
C——0  
A——10  
B——111  
D——110



$$WPL = 80 \times 1 + 10 \times 2 + 2 \times 3 + 8 \times 3 = 130$$

CAAABD: 0101010111110

C——0  
A——1  
B——111  
D——110



非前缀码解  
码有歧义

CAAABD: 0111111110

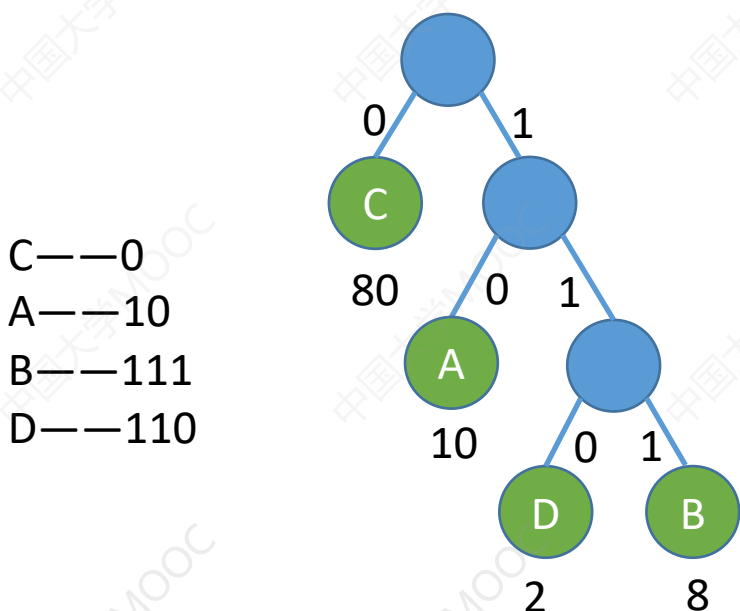
# 哈夫曼编码

固定长度编码——每个字符用相等长度的二进制位表示

可变长度编码——允许对不同字符用不等长的二进制位表示

若没有一个编码是另一个编码的前缀，则称这样的编码为前缀编码

有哈夫曼树得到哈夫曼编码——字符集中的每个字符作为一个叶子结点，各个字符出现的频度作为结点的权值，根据之前介绍的方法构造哈夫曼树



$$WPL = 80 \times 1 + 10 \times 2 + 2 \times 3 + 8 \times 3 = 130$$

哈夫曼树不唯一，因此哈夫曼编码不唯一

C——1  
A——00  
B——011  
D——010

$$WPL = 80 \times 1 + 10 \times 2 + 2 \times 3 + 8 \times 3 = 130$$

哈夫曼编码可用于数据压缩



## 英文字母频次



英文字母使用频率表:(%)

A 8.19	B 1.47	C 3.83	D 3.91	E 12.25	F 2.26	G 1.71
H 4.57	I 7.10	J 0.14	K 0.41	L 3.77	M 3.34	N 7.06
O 7.26	P 2.89	Q 0.09	R 6.85	S 6.36	T 9.41	
U 2.58	V 1.09	W 1.59	X 0.21	Y 1.58	Z 0.08	

试试设计哈夫曼编码，并计算数据压缩率

# 知识回顾与重要考点

## 哈夫曼树

### 概念

结点的权：某种特定含义的数值

结点的带权路径长度 = 根到结点路径长度\*结点的权值

树的带权路径长度(WPL) = 树中所有叶子结点的带权路径长度之和

哈夫曼树（最优二叉树）：在含有给定的n个带权叶结点的二叉树中，WPL 最小的二叉树

### 构造哈夫曼树

每次选两个根节点权值最小的树合并，并将二者权值之和作为新的根节点的权值

哈夫曼树不唯一，但WPL必然都是最小值

### 哈夫曼编码

将字符频次作为字符结点权值，构造哈夫曼树，即可得哈夫曼编码，可用于数据压缩

前缀编码——没有一个编码是另一个编码的前缀

固定长度编码——每个字符用相等长度的二进制位表示

可变长度编码——允许对不同字符用不等长的二进制位表示

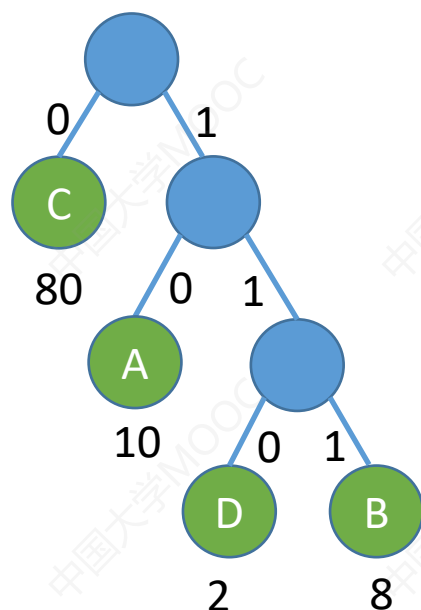
# 结局

可变长度编码——允许对不同字符用不等长的二进制位表示

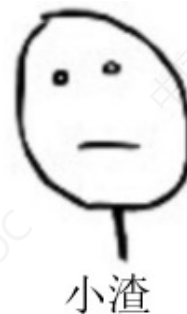
若没有一个编码是另一个编码的前缀，则称这样的编码为前缀编码

C—0  
A—10  
B—111  
D—110

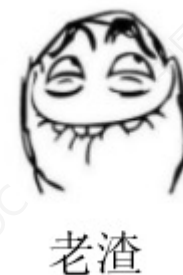
哈夫曼编码



$$WPL = 80 \times 1 + 10 \times 2 + 2 \times 3 + 8 \times 3 = 130 \text{ bit}$$



100道选择题



# 欢迎大家对本节视频进行评价~



学员评分：5.5.1 哈夫曼树

扫一扫二维码打开或分享给好友



— 腾讯文档 —

可多人实时在线编辑，权限安全可控



公众号：王道在线



b站：王道计算机教育



抖音：王道计算机考研