



举个栗子



西南大学附属中学
High School Affiliated to Southwest University

在很多问题的处理过程中，需要进行大量的条件判断，这些判断结构的设计直接影响着程序的执行效率。
例如，编制一个程序，将百分制转换成五个等级输出。



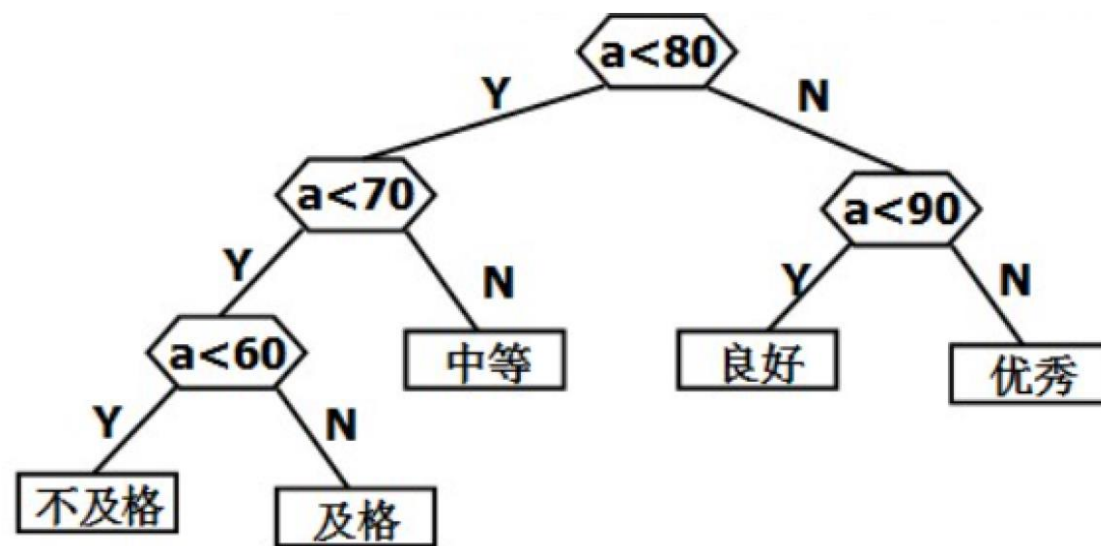
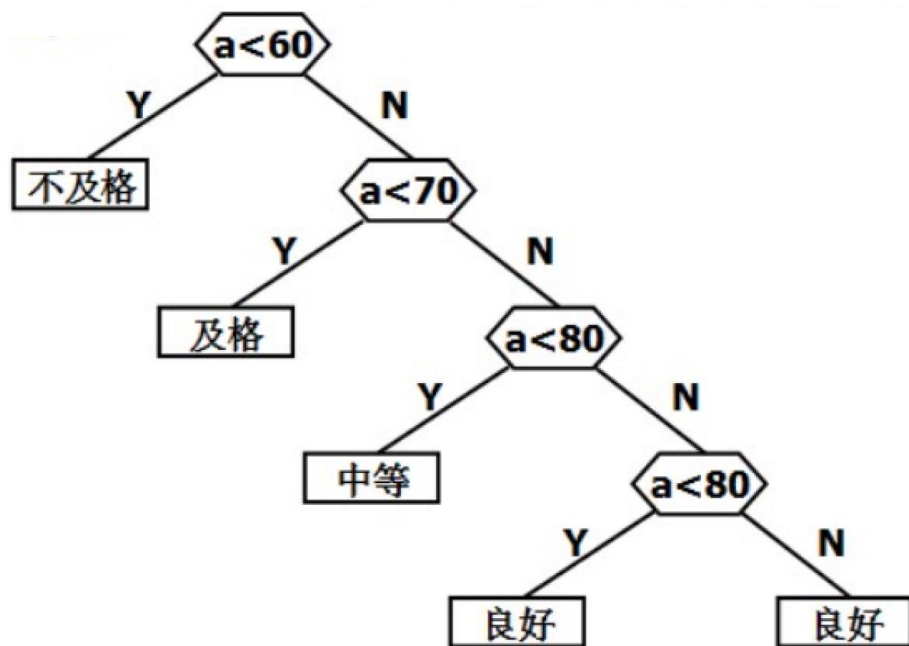


```
if(score<60)
    cout<<"Bad"<<endl;
else if(score<70)
    cout<<"Pass"<<endl;
else if(score<80)
    cout<<"General"<<endl;
else if(score<90)
    cout<<"Good"<<endl;
else
    cout<<"Very good!"<<endl;
```

在实际中，学生成绩呈正态分布，
也就是“中间多，两头少”，
在五个等级上的分布是不均匀的。

需要反复调用

分数	0~59	60~69	70~79	80~89	90~100
比例	0.05	0.15	0.40	0.30	0.10



判定过程最优的二叉树为哈夫曼树



- 1、初始化： 根据给定的 n 个权值 $\{w_1, w_2, \dots, w_n\}$ 构成 n 棵二叉树的集合 $F=\{T_1, T_2, \dots, T_n\}$ ，其中每棵二叉树 T_i 中只有一个带权 w_i 的根结点，左右子树均空；
- 2、 找最小树：在 F 中选择两棵根结点权值最小的树作为左右子树构造一棵新的二叉树，且至新的二叉树的根结点的权值为其左右子树上根结点的权值之和；
- 3、删除与加入：在 F 中删除这两棵树，并将新的二叉树加入 F 中；
- 4、判断：重复前两步（2和3），直到 F 中只含有一棵树为止。

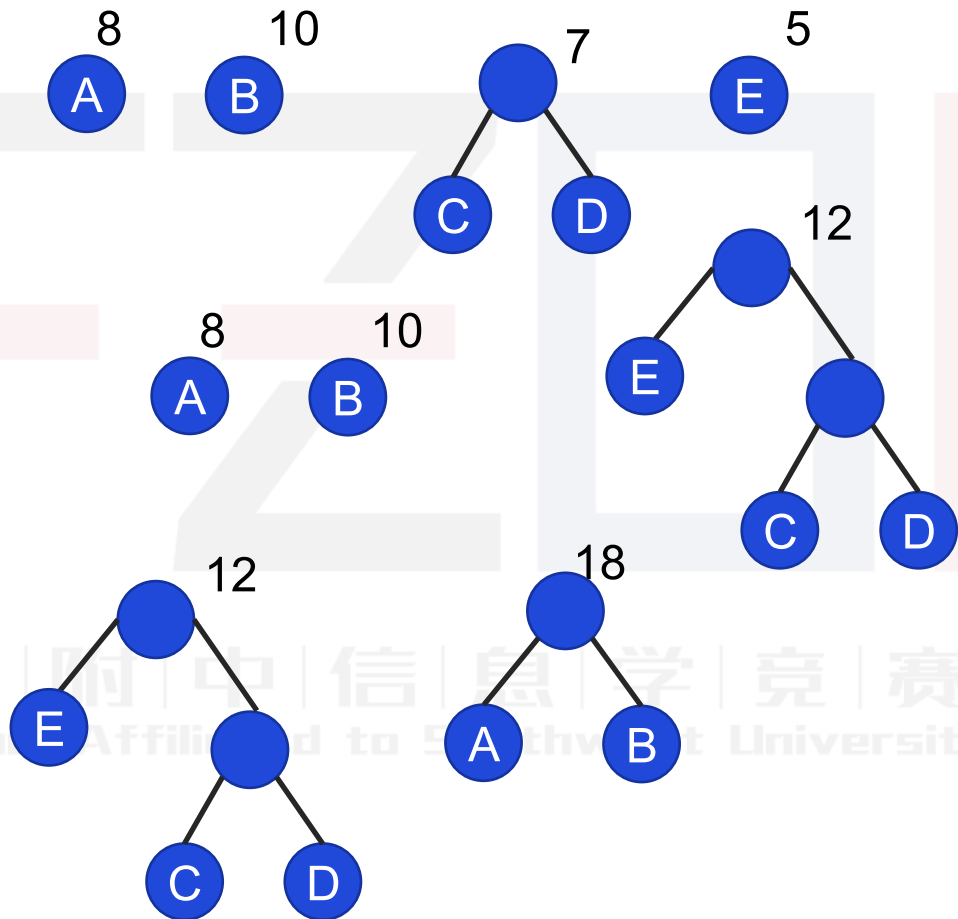


现有一个由5个不同符号组成的30个符号的字符串：
BABACAC ADADABB CBABEBE DDABEEEEBB

1 首先计算出每个字符出现的次数（概率）：

字符	次数
A	8
B	10
C	3
D	4
E	5

2 不断从中选出权值（次数）最小的两个，作为左右子树构造一棵二叉树，根节点权值为子树根节点权值和



关键：每次都选取最小的两个值合并成为一个新的结点



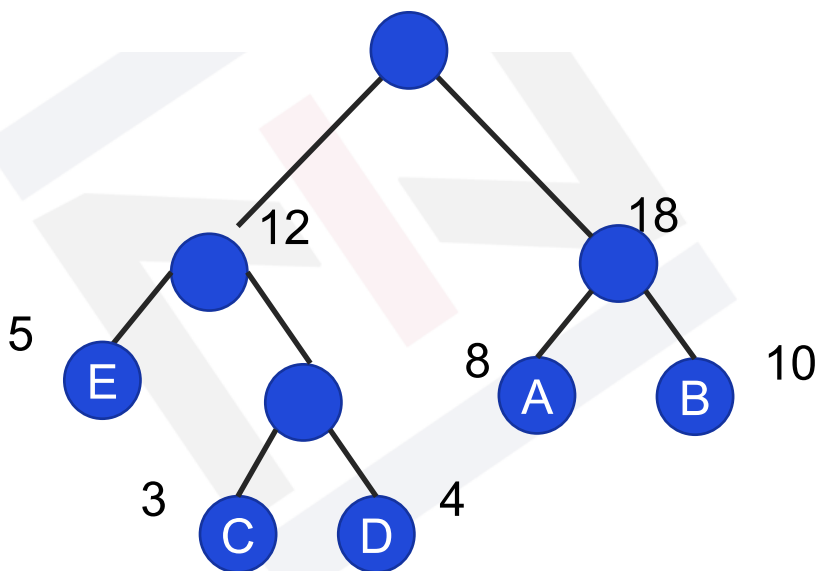
带权路径长度 (WPL)



西南大学附属中学
High School Affiliated to Southwest University

Weighted Path Length: 设二叉树有 n 个叶子结点, 每个叶子结点带有权值 w_k , 从根结点到每个叶子结点的长度为 l_k , 则每个叶子结点的带权路径长度之和就是:

$$WPL = \sum_{k=1}^n w_k l_k$$



西 | 大 | 附 | 中 | 信 | 息 | 学 | 竞 | 赛 |
High School Affiliated to Southwest University

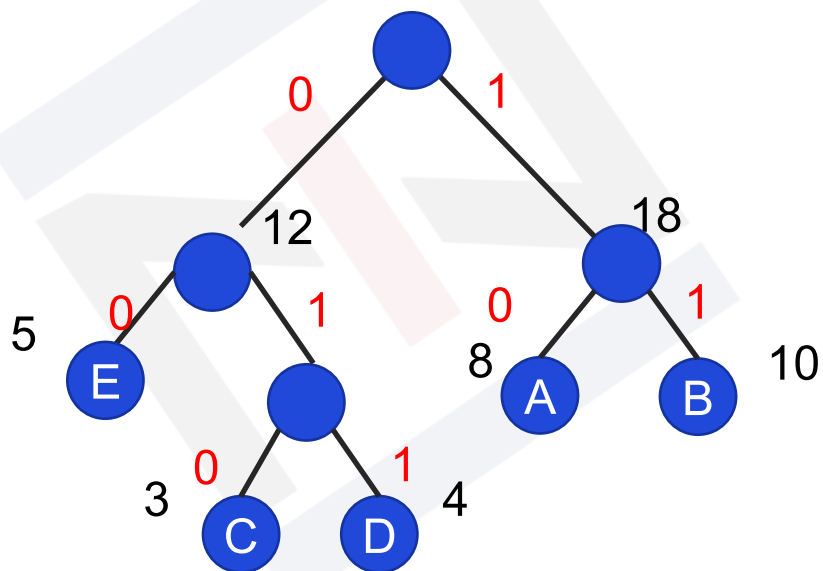


哈夫曼编码的生成



西南大学附属中学
High School Affiliated to Southwest University

对于每个节点，取左子树编码为“0”，右子树编码为“1”



A(10) B(11) C(010) D(011) E(00)

这样就可以使最终的编码长度达到最短进而达到压缩的目的，因为出现次数最多的字母的编码长度最短，出现次数最少的字母编码较长。

在计算机数据处理中，哈夫曼编码使用变长编码表对源符号进行编码，出现频率较高的源符号采用较短的编码，出现频率较低的符号采用较长的编码，使编码之后的字符串字符串的平均长度、期望值降低，以达到无损压缩数据的目的。



哈夫曼树"每次取最小两值合并为新的结点"这一思想可以通过**小根堆**实现。

小根堆每次弹出两个值，然后将二者的和再插入小根堆中。

```
#include<stdio.h>
#include<queue>
using namespace std;
int n,ans,k;
priority_queue<int,vector<int>,greater<int> > q;//小根堆
int main(){
    scanf("%d",&n);
    for(int i=1;i<=n;i++){
        scanf("%d",&k);
        q.push(k);
    }
    for(int i=1;i<n;i++){
        int a=q.top();
        q.pop();
        int b=q.top();
        q.pop();
        ans+=a+b;
        q.push(a+b);
    }
    printf("%d",ans);
    return 0;
}
```




https://wardseptember.blog.csdn.net/article/details/80717653?spm=1001.2101.3001.6650.2&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-2-80717653-blog-124295271.pc_relevant_3mothn_strategy_and_data_recovery&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-2-80717653-blog-124295271.pc_relevant_3mothn_strategy_and_data_recovery&utm_relevant_index=3



在哈夫曼编码中,若编码长度只允许小于等于4,则除了已对两个字符编码为0和10外,还可以最多对几个字符编码?

是4个: 1100,1101,1110,1111

n个字符的哈夫曼编码中,各字符编码长度最大值为?

B

A、n B、n-1 C、不大于n/2的最大的整数 D、大于n/2的最小的整数。

下列关于Huffman编码的描述, 其中“ 不正确” 的一个是:

A.赫夫曼编码是一种前缀编码

B.赫夫曼编码是一种等长编码

C.采用赫夫曼编码使得编码后的报文二进制串总长度尽可能短

D.在赫夫曼编码中, 任意一个字符的编码都不可能是另一字符编码的前缀

B



下列关于Huffman编码的描述，其中“不正确”的一个是：

- A. 是一种不等长编码方式
- B. 是一种无失真的编码方式
- C. 得到的编码是唯一的
- D. 是最佳编码方式之一

C