## 编写哈夫曼编码

有如下字符串内容：

AAABABC

1. 首先统计字母出现次数，对出现次数进行升序排列：1 2 4。
2. 构造哈夫曼树。

   从序列中取两个最小值，求和做根

   把刚刚产生的根替换回原序列


将树的路径用二进制数字填充

从根结点出发，到达叶子结点的路径有3条：

4: 1 → A

2: 0 1 → B

1: 0 0 → C

此为A、B、C的哈夫曼编码

## 编写哈夫曼编码

有如下字符串内容：

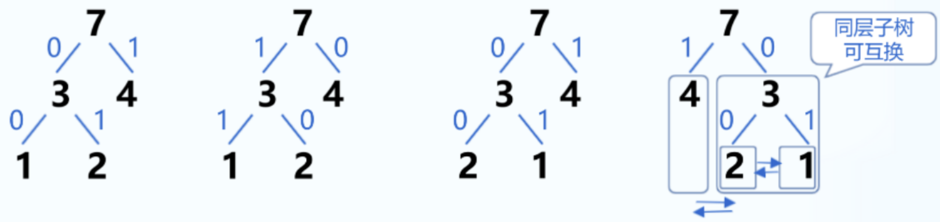
AAABABC

1. 首先统计字母出现次数，对出现次数进行升序排列：1 2 4。
2. 构造哈夫曼树。

   从序列中取两个最小值，求和做根

   把刚刚产生的根替换回原序列

交换完依然要满足哈夫曼树的构造要求。




## 图的概念

图是一种数据结构。

1. 概念：图是由有限个结点和结点之间的边组成。通常表示为G=(V,E)。
2. 两种图：（1）无向图：图中任意一条边都没有方向。

   （2）有向图：图中任意一条边都有方向。
3. 简单图：若同一条边不重复出现，且不存在结点到自身的边，则称图为简单图。
4. 完全图：（1）有向完全图：有向图任意两个结点都存在方向互为相反的两条边。

   （2）无向完全图：无向图任意两个结点都存在边。

   n个结点: 有向图边数=n*(n-1)

   无向图边数=n*(n-1)/2
5. 结点v的度：结点v相连边的数目。

   （1）结点v入度：有向图中以结点v作为终点边的数目。

   （2）结点v出度：有向图中以结点v作为起点边的数目。

   有向图中，结点的度=入度+出度。

## 图的概念

1. 权和网：

（1）图中边上的数字叫做“权值”。

（2）带权的图通常称为“网”。

注意：权值可以表示从一个结点到另一个结点的**距离**，**时间**或其他一些**代价**。

2. 路径和回路：

（1）从一个结点到另一个结点，途中经过的所有结点称为一条路径。

（2）一条路径的起点与终点相同，则称此路径为“回路”。

3. 子图： 图G1所有的结点和边都属于图G2，则称图G1为图G2的子图。

G1=(V1,E1) G2=(V2,E2) 如果V1是V2子集，且E1是E2子集。

则G1是G2的子图。

## 图的概念

4. 连通与连通图：

（1）无向图中，两个结点间存在路径，则称两个结点连通。

（2）无向图中，任意两个结点都连通，则称该图为连通图。

5. 强连通与强连通图：

（1）有向图中，两个结点相互可以到达，则称两个结点强连通。

（2）有向图中，任意两个结点间都强连通，则称有向图是强连通图。

6. 强连通分量： 有向图中，最大的强连通子图。

## 图的存储

1. 邻接矩阵：存储图的二维数组。

矩阵行列规模要大于最大结点编号

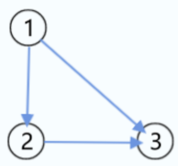
（1）图：用**1**表示两个结点有边；用**0**表示两个结点没边。

（2）网：用**权值**表示两个结点有边；用**极大值**表示两个结点没边。

极大值要大于最大的权值

相同点：有向边赋1次；无向边赋2次。

以有向图为例：




| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | **1** | **1** |
| 2 | 0 | 0 | 0 | **1** |
| 3 | 0 | 0 | 0 | 0 |

g[4][4]

## 排列

排列：从n个不同的元素中，任取m(m≤n)个元素，按照一定的顺序排成一列，叫做从n个不同元素中取出m个元素的一个排列。

当m等于n时，表示在n个不同元素中，所有元素进行排列，称为全排列。

排列数：从n个不同元素中，任取 m个元素的所有排列的个数叫做从n个元素中取出m个元素的排列数，用 $A_n^m$ 表示。

$A_5^3$　　表示从5个不同元素中取出3个的排列数

$A_{10}^2$　　表示从10个不同元素中取出2个的排列数

## 排列数的计算公式

$A_n^m$　表示从n个不同元素中取出m个的排列数

| 第1步：确定第一个 | → | 第2步：确定第二个 | → | 第3步：确定第三个 | …… | → | 第m步：确定第m个 |
|---|---|---|---|---|---|---|---|
| n种方法 | | n-1种方法 | | n-2种方法 | …… | | n-m+1种方法 |

利用乘法原理，总排列数为：n x (n-1) x (n-2) x...x (n-m+1)

$A_n^m$ 计算公式：n x (n-1) x (n-2) x...x (n-m+1)

计算规则：

从数字n开始乘，之后的每个数字递减1，乘到n-m+1结束，共m个数字相乘。

## 组合

组合：从n个不同的元素中，任取m(m≤n)个元素，并成一组，叫做从n个不同元素中取出m个元素的一个组合。

注意：如果两个组合中的元素完全相同，不管元素的顺序如何，都是相同的组合。

| 排列 | VS | 组合 |
|---|---|---|
| 有顺序 | | 无顺序 |

组合数：从n个不同元素中取出m个元素的所有组合的个数，用$C_n^m$ 表示。

## 组合数的计算公式

排序：从n个不同元素中选出m个元素，进行排序。
而组合是从n个不同元素中选出m个元素，不进行排序。

所以计算排列数时，都可以分解为以下两步：

分解步骤：
1. 先从n个元素中选出m个(组合) → $C_n^m$
2. 对选出的m个元素，进行全排列(排列) → $A_m^m$

从而得出： $A_n^m = C_n^m \times A_m^m \Rightarrow C_n^m = \frac{A_n^m}{A_m^m}$

另外注意，组合数中规定：$C_n^0 = 1$

## 常用解题方法

| 常用方法 | 方法介绍 | 常见问题 |
|---|---|---|
| 捆绑法 | 把相邻的若干特殊元素捆绑为一个元素，然后与其他元素全排列，然后再松绑，将这些特殊元素做全排列。 | 相邻问题 |
| 插空法 | 即先安排好没有限制条件的元素，然后将没有限制条件的元素按要求插入到排好的元素之间。 | 不相邻问题 |
| 隔板法 | 将n个相同元素有序地分成几组，且每组至少有一个元素，则在n-1个空中可插入m个板，可以把n个元素分成m+1组 | 元素分组问题 |
| 逆向法 | 首先计算所有可能的情况数（即总情况数），然后找出不满足题目要求的情况数（即反面情况数），最后通过总情况数减去反面情况数来得到满足题目要求的情况数。 | 正向计算困难的问题 |