

引入

排列组合是组合数学中的基础。排列就是指从给定个数的元素中取出指定个数的元素进行排序；组合则是指从给定个数的元素中仅仅取出指定个数的元素，不考虑排序。排列组合的中心问题是研究给定要求的排列和组合可能出现的情况总数。排列组合与古典概率论关系密切。

在高中初等数学中，排列组合多是利用列表、枚举等方法解题。

加法 & 乘法原理

加法原理

完成一个工程可以有 n 类办法， $a_i (1 \leq i \leq n)$ 代表第 i 类方法的数目。那么完成这件事共有 $S = a_1 + a_2 + \cdots + a_n$ 种不同的方法。

乘法原理

完成一个工程需要分 n 个步骤， $a_i (1 \leq i \leq n)$ 代表第 i 个步骤的不同方法数目。那么完成这件事共有 $S = a_1 \times a_2 \times \cdots \times a_n$ 种不同的方法。

排列与组合基础

排列数

从 n 个不同元素中，任取 m ($m \leq n$, m 与 n 均为自然数，下同) 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；从 n 个不同元素中取出 m ($m \leq n$) 个元素的所有排列的个数，叫做从 n 个不同元素中取出 m 个元素的排列数，用符号 A_n^m (或者是 P_n^m) 表示。

排列的计算公式如下：

$$A_n^m = n(n-1)(n-2) \cdots (n-m+1) = \frac{n!}{(n-m)!}$$

$n!$ 代表 n 的阶乘，即 $6! = 1 \times 2 \times 3 \times 4 \times 5 \times 6$ 。

公式可以这样理解： n 个人选 m 个来排队 ($m \leq n$)。第一个位置可以选 n 个，第二位置可以选 $n-1$ 个，以此类推，第 m 个 (最后一个) 可以选 $n-m+1$ 个，得：

$$A_n^m = n(n-1)(n-2) \cdots (n-m+1) = \frac{n!}{(n-m)!}$$

全排列： n 个人全部来排队，队长为 n 。第一个位置可以选 n 个，第二位置可以选 $n-1$ 个，以此类推得：

$$A_n^n = n(n-1)(n-2) \cdots 3 \times 2 \times 1 = n!$$

全排列是排列数的一个特殊情况。

组合数

从 n 个不同元素中，任取 $m \leq n$ 个元素组成一个集合，叫做从 n 个不同元素中取出 m 个元素的一个组合；从 n 个不同元素中取出 $m \leq n$ 个元素的所有组合的个数，叫做从 n 个不同元素中取出 m 个元素的组合数，用符号 $\binom{n}{m}$ 来表示，读作「 n 选 m 」。

组合数计算公式

$$\binom{n}{m} = \frac{A_n^m}{m!} = \frac{n!}{m!(n-m)!}$$

如何理解上述公式？我们考虑 n 个人选 m 个出来 ($m \leq n$)，不排队，不在乎顺序。如果在乎顺序那么就是 A_n^m ，如果不在乎那么就要除掉重复，那么重复了多少？同样选出来的 m 个人，他们还要「全排」得 $m!$ ，所以得：

$$\begin{aligned}\binom{n}{m} \times m! &= A_n^m \\ \binom{n}{m} &= \frac{A_n^m}{m!} = \frac{n!}{m!(n-m)!}\end{aligned}$$

组合数也常用 C_n^m 表示，即 $C_n^m = \binom{n}{m}$ 。现在数学界普遍采用 $\binom{n}{m}$ 的记号而非 C_n^m 。

组合数也被称为「二项式系数」，下文二项式定理将会阐述其中的联系。

特别地，规定当 $m > n$ 时， $A_n^m = \binom{n}{m} = 0$ 。

插板法

插板法 (Stars and bars) 是用于求一类给相同元素分组的方案数的一种技巧，也可以用于求一类线性不定方程的解的组数。

正整数和的数目

问题一：现有 n 个 **完全相同** 的元素，要求将其分为 k 组，保证每组至少有一个元素，一共有多少种分法？

考虑拿 $k-1$ 块板子插入到 n 个元素两两形成的 $n-1$ 个空里面。

因为元素是完全相同的，所以答案就是 $\binom{n-1}{k-1}$ 。

本质是求 $x_1 + x_2 + \cdots + x_k = n$ 的正整数解的组数。

非负整数和的数目

问题二：如果问题变化一下，每组允许为空呢？

显然此时没法直接插板了，因为有可能出现很多块板子插到一个空里面的情况，非常不好计算。

我们考虑创造条件转化成有限制的问题一，先借 k 个元素过来，在这 $n+k$ 个元素形成的 $n+k-1$ 个空里面插板，答案为

$$\binom{n+k-1}{k-1} = \binom{n+k-1}{n}$$

虽然不是直接求的原问题，但这个式子就是原问题的答案，可以这么理解：

开头我们借来了 k 个元素，用于保证每组至少有一个元素，插完板之后再把这 k 个借来的元素从 k 组里面拿走。因为元素是相同的，所以转化过的情况和转化前的情况可以一一对应，答案也就是相等的。

由此可以推导出插板法的公式： $\binom{n+k-1}{n}$ 。

本质是求 $x_1 + x_2 + \cdots + x_k = n$ 的非负整数解的组数（即要求 $x_i \geq 0$ ）。

不同下界整数和的数目

问题三：如果再扩展一步，要求对于第 i 组，至少要分到 a_i , $\sum a_i \leq n$ 个元素呢？

本质是求 $x_1 + x_2 + \cdots + x_k = n$ 的解的数目，其中 $x_i \geq a_i$ 。

类比无限制的情况，我们借 $\sum a_i$ 个元素过来，保证第 i 组至少能分到 a_i 个。也就是令

$$x'_i = x_i - a_i$$

得到新方程：

$$\begin{aligned}(x'_1 + a_1) + (x'_2 + a_2) + \cdots + (x'_k + a_k) &= n \\ x'_1 + x'_2 + \cdots + x'_k &= n - a_1 - a_2 - \cdots - a_k \\ x'_1 + x'_2 + \cdots + x'_k &= n - \sum a_i\end{aligned}$$

其中

$$x'_i \geq 0$$

然后问题三就转化成了问题二，直接用插板法公式得到答案为

$$\binom{n - \sum a_i + k - 1}{n - \sum a_i}$$

不相邻的排列

$1 \sim n$ 这 n 个自然数中选 k 个，这 k 个数中任何两个数都不相邻的组合有 $\binom{n - k + 1}{k}$ 种。

二项式定理

在进入排列组合进阶篇之前，我们先介绍一个与组合数密切相关的定理——二项式定理。

二项式定理阐明了一个展开式的系数：

$$(a + b)^n = \sum_{i=0}^n \binom{n}{i} a^{n-i} b^i$$

证明可以采用数学归纳法，利用 $\binom{n}{k} + \binom{n}{k-1} = \binom{n+1}{k}$ 做归纳。

二项式定理也可以很容易扩展为多项式的形式：

设 n 为正整数， x_i 为实数，

$$(x_1 + x_2 + \cdots + x_t)^n = \sum_{\text{满足 } n_1 + \cdots + n_t = n \text{ 的非负整数解}} \binom{n}{n_1, n_2, \dots, n_t} x_1^{n_1} x_2^{n_2} \cdots x_t^{n_t}$$

其中的 $\binom{n}{n_1, n_2, \dots, n_t}$ 是多项式系数，它的性质也很相似：

$$\sum \binom{n}{n_1, n_2, \dots, n_t} = t^n$$

排列与组合进阶篇

接下来我们介绍一些排列组合的变种。

多重集的排列数 | 多重组合数

请大家一定要区分 **多重组合数** 与 **多重集的排列数**！两者是完全不同的概念！

多重集是指包含重复元素的广义集合。设 $S = \{n_1 \cdot a_1, n_2 \cdot a_2, \dots, n_k \cdot a_k\}$ 表示由 n_1 个 a_1 , n_2 个 a_2 , ..., n_k 个 a_k 组成的多重集, S 的全排列个数为

$$\frac{n!}{\prod_{i=1}^k n_i!} = \frac{n!}{n_1!n_2!\cdots n_k!}$$

相当于把相同元素的排列数除掉了。具体地, 你可以认为你有 k 种不一样的球, 每种球的个数分别是 n_1, n_2, \dots, n_k , 且 $n = n_1 + n_2 + \dots + n_k$ 。这 n 个球的全排列数就是 **多重集的排列数**。多重集的排列数常被称作 **多重组合数**。我们可以用多重组合数的符号表示上式:

$$\binom{n}{n_1, n_2, \dots, n_k} = \frac{n!}{\prod_{i=1}^k n_i!}$$

可以看出, $\binom{n}{m}$ 等价于 $\binom{n}{m, n-m}$, 只不过后者较为繁琐, 因而不采用。

多重集的组合数 1

设 $S = \{n_1 \cdot a_1, n_2 \cdot a_2, \dots, n_k \cdot a_k\}$ 表示由 n_1 个 a_1 , n_2 个 a_2 , ..., n_k 个 a_k 组成的多重集。那么对于整数 $r (r < n_i, \forall i \in [1, k])$, 从 S 中选择 r 个元素组成一个多重集的方案数就是 **多重集的组合数**。这个问题等价于 $x_1 + x_2 + \dots + x_k = r$ 的非负整数解的数目, 可以用插板法解决, 答案为

$$\binom{r+k-1}{k-1}$$

多重集的组合数 2

考虑这个问题: 设 $S = \{n_1 \cdot a_1, n_2 \cdot a_2, \dots, n_k \cdot a_k\}$ 表示由 n_1 个 a_1 , n_2 个 a_2 , ..., n_k 个 a_k 组成的多重集。那么对于正整数 r , 从 S 中选择 r 个元素组成一个多重集的方案数。

这样就限制了每种元素的取的个数。同样的, 我们可以把这个问题转化为带限制的线性方程求解:

$$\forall i \in [1, k], x_i \leq n_i, \sum_{i=1}^k x_i = r$$

于是很自然地想到了容斥原理。容斥的模型如下:

1. 全集: $\sum_{i=1}^k x_i = r$ 的非负整数解。

2. 属性: $x_i \leq n_i$ 。

于是设满足属性 i 的集合是 S_i , $\overline{S_i}$ 表示不满足属性 i 的集合, 即满足 $x_i \geq n_i + 1$ 的集合 (转化为上面插板法的问题三)。那么答案即为

$$\left| \bigcap_{i=1}^k S_i \right| = |U| - \left| \bigcup_{i=1}^k \overline{S_i} \right|$$

根据容斥原理, 有:

$$\begin{aligned}
\left| \bigcup_{i=1}^k \overline{S_i} \right| &= \sum_i |\overline{S_i}| - \sum_{i,j} |\overline{S_i} \cap \overline{S_j}| + \sum_{i,j,k} |\overline{S_i} \cap \overline{S_j} \cap \overline{S_k}| - \dots \\
&\quad + (-1)^{k-1} \left| \bigcap_{i=1}^k \overline{S_i} \right| \\
&= \sum_i \binom{k+r-n_i-2}{k-1} - \sum_{i,j} \binom{k+r-n_i-n_j-3}{k-1} + \sum_{i,j,k} \binom{k+r-n_i-n_j-n_k-4}{k-1} - \dots \\
&\quad + (-1)^{k-1} \binom{k+r-\sum_{i=1}^k n_i-k-1}{k-1}
\end{aligned}$$

拿全集 $|U| = \binom{k+r-1}{k-1}$ 减去上式，得到多重集的组合数

$$Ans = \sum_{p=0}^k (-1)^p \sum_A \binom{k+r-1-\sum_A n_{A_i}-p}{k-1}$$

其中 A 是充当枚举子集的作用，满足 $|A| = p$, $A_i < A_{i+1}$ 。

圆排列

n 个人全部来围成一圈，所有的排列数记为 Q_n^n 。考虑其中已经排好的一圈，从不同位置断开，又变成不同的队列。

所以有

$$Q_n^n \times n = A_n^n \implies Q_n = \frac{A_n^n}{n} = (n-1)!$$

由此可知部分圆排列的公式：

$$Q_n^r = \frac{A_n^r}{r} = \frac{n!}{r \times (n-r)!}$$

组合数性质 | 二项式推论

由于组合数在 OI 中十分重要，因此在此介绍一些组合数的性质。

$$\binom{n}{m} = \binom{n}{n-m} \quad (1)$$

相当于将选出的集合对全集取补集，故数值不变。（对称性）

$$\binom{n}{k} = \frac{n}{k} \binom{n-1}{k-1} \quad (2)$$

由定义导出的递推式。

$$\binom{n}{m} = \binom{n-1}{m} + \binom{n-1}{m-1} \quad (3)$$

组合数的递推式（杨辉三角的公式表达）。我们可以利用这个式子，在 $O(n^2)$ 的复杂度下推导组合数。

$$\binom{n}{0} + \binom{n}{1} + \dots + \binom{n}{n} = \sum_{i=0}^n \binom{n}{i} = 2^n \quad (4)$$

这是二项式定理的特殊情况。取 $a = b = 1$ 就得到上式。

$$\sum_{i=0}^n (-1)^i \binom{n}{i} = [n=0] \quad (5)$$

二项式定理的另一种特殊情况，可取 $a = 1, b = -1$ 。式子的特殊情况是取 $n = 0$ 时答案为 1。

$$\sum_{i=0}^m \binom{n}{i} \binom{m}{m-i} = \binom{m+n}{m} \quad (n \geq m) \quad (6)$$

拆组合数的式子，在处理某些数据结构题时会用到。

$$\sum_{i=0}^n \binom{n}{i}^2 = \binom{2n}{n} \quad (7)$$

这是 (6) 的特殊情况，取 $n = m$ 即可。

$$\sum_{i=0}^n i \binom{n}{i} = n 2^{n-1} \quad (8)$$

带权和的一个式子，通过对 (3) 对应的多项式函数求导可以得证。

$$\sum_{i=0}^n i^2 \binom{n}{i} = n(n+1) 2^{n-2} \quad (9)$$

与上式类似，可以通过对多项式函数求导证明。

$$\sum_{l=0}^n \binom{l}{k} = \binom{n+1}{k+1} \quad (10)$$

通过组合分析——考虑 $S = \{a_1, a_2, \dots, a_{n+1}\}$ 的 $k+1$ 子集数可以得证，在恒等式证明中比较常用。

$$\binom{n}{r} \binom{r}{k} = \binom{n}{k} \binom{n-k}{r-k} \quad (11)$$

通过定义可以证明。

$$\sum_{i=0}^n \binom{n-i}{i} = F_{n+1} \quad (12)$$

其中 F 是斐波那契数列。

二项式反演

记 f_n 表示恰好使用 n 个不同元素形成特定结构的方案数， g_n 表示从 n 个不同元素中选出 $i \geq 0$ 个元素形成特定结构的总方案数。

若已知 f_n 求 g_n ，那么显然有：

$$g_n = \sum_{i=0}^n \binom{n}{i} f_i$$

若已知 g_n 求 f_n ，那么：

$$f_n = \sum_{i=0}^n \binom{n}{i} (-1)^{n-i} g_i$$

上述已知 g_n 求 f_n 的过程，就称为 **二项式反演**。

证明

将反演公式的 g_i 展开得到：

$$\begin{aligned} f_n &= \sum_{i=0}^n \binom{n}{i} (-1)^{n-i} \left[\sum_{j=0}^i \binom{i}{j} f_j \right] \\ &= \sum_{i=0}^n \sum_{j=0}^i \binom{n}{i} \binom{i}{j} (-1)^{n-i} f_j \end{aligned}$$

先枚举 j ，再枚举 i ，得到：

$$\begin{aligned} f_n &= \sum_{j=0}^n \sum_{i=j}^n \binom{n}{i} \binom{i}{j} (-1)^{n-i} f_j \\ &= \sum_{j=0}^n f_j \sum_{i=j}^n \binom{n}{i} \binom{i}{j} (-1)^{n-i} \end{aligned}$$

使用 [「组合数性质」](#) [二项式推论」](#) 的公式 (11) 得到：

$$\begin{aligned} f_n &= \sum_{j=0}^n f_j \sum_{i=j}^n \binom{n}{i} \binom{n-j}{i-j} (-1)^{n-i} \\ &= \sum_{j=0}^n \binom{n}{j} f_j \sum_{i=j}^n \binom{n-j}{i-j} (-1)^{n-i} \end{aligned}$$

令 $k = i - j$ 。则 $i = k + j$ ，上式转换为：

$$f_n = \sum_{j=0}^n \binom{n}{j} f_j \sum_{k=0}^{n-j} \binom{n-j}{k} (-1)^{n-j-k} 1^k$$

使用 [「组合数性质」](#) [二项式推论」](#) 的公式 (5) 得到：

$$f_n = \sum_{j=0}^n \binom{n}{j} f_j [n = j] = f_n$$

证毕。