XJOI 进阶讲义（二）

容斥

Mr_Spade

2021.9

概述

容斥和动态规划是 OI 的计数题中应用最广泛、思维难度上限最高的知识点。几乎所有的计数难题中，容斥或动态规划都会作为难点出现。因此，比较好地掌握这两个知识点是一个成熟的算法竞赛选手应当具备的能力。

和动态规划相比，容斥的一个隐含的难点在于其定义并不清晰，应用到题目中的方式也非常多样，难以用统一的形式描述。

本讲主要先引出容斥的常用方法并分析它们之间的联系，再列举几道灵活运用容斥（主要是容斥原理）的题目，从中分析归纳一种较为通用的容斥方法。在进行容斥时，最重要的是对量进行清晰的定义，用严谨的方法完成每个解题步骤，这样我们才能思路清晰地用容斥解决更多问题。

定义

在计数问题中，我们通常需要统计所有合法方案的数目或带权和。如果我们利用加法原理和乘法原理，不重不漏地对所有合法方案进行了统计，那么这样的方法称为直接计数法。（一个不太准确但非常直观的理解方式是，只用加法和乘法进行统计的方法叫做直接计数法。）

除直接计数法以外的计数方法则统称为间接计数法，间接计数法通常会先进行重复或遗漏的带权计数，但将计数结果全部统计后，所有重叠或遗漏的部分会在系数上相互抵消，因此仍能得到准确的答案。

定义

在计数问题中，我们通常需要统计所有合法方案的数目或带权和。如果我们利用加法原理和乘法原理，不重不漏地对所有合法方案进行了统计，那么这样的方法称为直接计数法。（一个不太准确但非常直观的理解方式是，只用加法和乘法进行统计的方法叫做直接计数法。）

除直接计数法以外的计数方法则统称为间接计数法，间接计数法通常会先进行重复或遗漏的带权计数，但将计数结果全部统计后，所有重叠或遗漏的部分会在系数上相互抵消，因此仍能得到准确的答案。

广义地讲，所有间接计数的方法都可以称作容斥。

定义

在计数问题中，我们通常需要统计所有合法方案的数目或带权和。如果我们利用加法原理和乘法原理，不重不漏地对所有合法方案进行了统计，那么这样的方法称为直接计数法。（一个不太准确但非常直观的理解方式是，只用加法和乘法进行统计的方法叫做直接计数法。）

除直接计数法以外的计数方法则统称为间接计数法，间接计数法通常会先进行重复或遗漏的带权计数，但将计数结果全部统计后，所有重叠或遗漏的部分会在系数上相互抵消，因此仍能得到准确的答案。

广义地讲，所有间接计数的方法都可以称作容斥。

所有间接计数方法中，应用最广、变式最多的则是狭义上的容斥——对子集的容斥，也就是容斥原理。本讲中主要对容斥原理进行讨论。

容斥原理

容斥原理的两个最基本的形式如下（顺带一提，这就是沃尔什变换对或卷积和与卷积进行的变换）：

$$f_S=\sum_{T\subseteq S}g_T \iff g_S=\sum_{T\subseteq S}(-1)^{|S|-|T|}f_T \tag{*}$$

$$f_S=\sum_{S\subseteq T\subseteq U}g_T \iff g_S=\sum_{S\subseteq T\subseteq U}(-1)^{|T|-|S|}f_T \tag{**}$$

在实际应用中，我们会根据题意赋予 f_S, g_S 以组合意义，则 $(*)(**)$ 式就变为可供利用的等式。例如，令全集 $U=\{1,2,\ldots,n\}$，f_S 代表长度为 n 的置换中，保持集合 S 内的元素不动的置换的数目，g_S 代表长度为 n 的置换中，**恰好**保持集合 S 内的元素不动的置换的数目，那么显然 $(**)$ 的左式成立。于是，我们推出右式也成立，从而可以用便于计算的 f 推出 g。

容斥原理

首先简单地进行证明，以第一个式子为例。考虑 $\sum_{T\subseteq S}\sum_{G\subseteq T}(-1)^{|T|-|G|} = \sum_{G\subseteq S}\sum_{G\subseteq T\subseteq S}(-1)^{|T|-|G|} = \sum_{G\subseteq S}\sum_{T\subseteq S\setminus G}(-1)^{|T|}$。

当 $S = G$ 时，$\sum_{T\subseteq S\setminus G}(-1)^{|T|}$ 显然为 1，否则，任取 $a \in S \setminus G$，则所有 $S \setminus G$ 的子集 T 可以和 $T\Delta\{a\}$ 唯一对应，且两者权值恰好抵消（这是上一讲的方法的再一次应用），于是 $\sum_{T\subseteq S\setminus G}(-1)^{|T|} = 0$。

综上，$\sum_{T\subseteq S\setminus G}(-1)^{|T|} = [S = G]$，于是显然 $f_S = \sum_{T\subseteq S}\sum_{G\subseteq T}(-1)^{|T|-|G|}f_G$，令 $g_S = \sum_{T\subseteq S}(-1)^{|S|-|T|}f_T$，我们就得到了 $(*)$ 式。

二项式反演

在容斥原理中，有一种特殊情况。如果在全集 U 中，所有元素的地位是等价的，那么 f_S 的值就只和 $|S|$ 有关。例如，之前所举的置换例子中，S 的各个元素地位就是等价的。此时，我们可以把 $f_{|S|}$ 简化为 f_i，其中 $i = |S|$。相应地，容斥原理的枚举子集可以改进为枚举子集的大小，再用组合数计算大小为特定值的子集的数目。此时 $(*)(**)$ 式被改造为如下：

$$f_i = \sum_{j=0}^{i} \binom{i}{j} g_j \iff g_i = \sum_{j=0}^{i} (-1)^{i-j} \binom{i}{j} f_j \tag{1}$$

$$f_i = \sum_{j=i}^{n} \binom{n-i}{j-i} g_j \iff g_i = \sum_{j=i}^{n} (-1)^{j-i} \binom{n-i}{j-i} f_j \tag{2}$$

这就是二项式反演。

使用之前的置换作为例子，显然有 $f_i = (n-i)!$，于是 $g_0 = \sum_{j=0}^{n} (-1)^j \binom{n}{j} (n-j)!$，这就是错排公式。

min-max 容斥

严格来说，min-max 容斥是一个形式上与容斥原理类似的公式，本质与容斥原理不同。不过，min-max 容斥是一种间接计数法，属于广义上的容斥。

min-max 容斥公式可以使用容斥原理推导。具体来说，假设对所有子集 T，有容斥系数 f_T，满足：

$$kth\max(S)=\sum_{T\subseteq S,T\neq\emptyset}f_T\min(T) \tag{1}$$

则此等式符合 $(*)$ 的左式，利用右式则有：

$$f_S=\frac{1}{\min(S)}\sum_{T\subseteq S,|T|\geq k}(-1)^{|S|-|T|}kth\max(T) \tag{2}$$

min-max 容斥

$$f_S=\frac{1}{\min(S)}\sum_{T\subseteq S,|T|\geq k}(-1)^{|S|-|T|}kth\max(T) \tag{3}$$

考虑计算右侧的和式，容易知当 $kth\max(T)$ 并非 S 中最小值 a 时，T 与 $T\Delta\{a\}$ 的 $kth\max$ 相同，贡献相互抵消（这又是上一讲的方法的再一次应用）。否则，有 $\binom{|S|-1}{k-1}$ 个集合包含 $\min(S)$，贡献均为 $(-1)^{|S|-k}\min(S)$，于是计算得 $f_S=(-1)^{|S|-k}\binom{|S|-1}{k-1}$。

综上，min-max 容斥公式为：

$$kth\max(S)=\sum_{T\subseteq S,T\neq\emptyset}(-1)^{|T|-k}\binom{|T|-1}{k-1}\min(T) \tag{4}$$

在之前的部分中，我们首先证明了容斥原理的公式，接着推导出了二项式反演（容斥原理的特殊情形），min-max 容斥（容斥原理推导出的实用公式）的结果。大家在利用容斥做题时，要积极寻找不同容斥方法之间的联系，不要完全将它们割裂开来，才更有利于理解容斥的本质。

接下来，将通过几道优质的容斥题目了解容斥原理究竟是如何应用在难题中的。

一道好题

有 n 种颜色的球，其中第 i 种颜色的有 a_i 个，现在希望将它们排成一列，使得相邻的球颜色不相同，求方案数。相同颜色的球视作完全相同的。

$m = \sum a_i \leq 10^5$

一道好题

由于每种颜色的球都有数目限制，因此直接计数法显然不太可行。转而考虑间接计数法。

间接计数法的一般步骤是：首先找到一类性质，满足所有符合这类性质的方案的带权和容易被整体计算（这是因为直接计数法通常是将整体划分为局部解决，当直接计数法遇到困难通常就意味着整体难以划分为局部，只能整体解决）；其次，判断所有能够被整体计算的结果能否通过赋予特定的权值再加和统计出原本的方案，这一步最常采用的方法是利用容斥原理确定权值。

相邻的球颜色不同可能性比较多，而相邻的球颜色相同似乎更好控制。是否能够对某一集合 $S \subseteq \{1, 2, \ldots, m-1\}$，统计“若 $i \in S$，则序列中的第 i 个球与第 $i+1$ 个球同色”的方案数？还是有些困难，因为相邻球是什么颜色还需要决定。

一道好题

那我们不妨将相邻球的颜色也确定。对于颜色 i，我们确定一个子集 $S_i \subseteq \{1,2,\ldots,a_i-1\}$，使得若 $j \in S_i$，则序列中的第 j 个颜色为 i 的球与第 $j+1$ 个颜色为 i 的球相邻。这样，就可以直接将颜色为 i 的球合并为 $a_i-|S_i|$ 段。相同颜色段顺序确定，不同颜色的段之间的顺序是任意的。

这就转化为一个经典问题了，先将所有段任意排列，再将每一种颜色的段按顺序排列，方案数就为 $\frac{\left(\sum_{i=1}^{n}(a_i-|S_i|)\right)!}{\prod_{i=1}^{n}(a_i-|S_i|)!}$。

第一步已经完成，现在考虑第二步：判断如何确定权值使得这些方案数的加权和为所求答案。考虑我们现在能够统计的是什么，就是 $S_1,S_2,\ldots,S_n$ 中的同色球对相邻的方案，那么我们设这是 $f_{S_1,S_2,\ldots,S_n}$，类比于错排的经验，我们令 $g_{S_1,S_2,\ldots,S_n}$ 为**恰好**是 $S_1,S_2,\ldots,S_n$ 中的同色球对相邻的方案，那么 $g_{\emptyset,\emptyset,\ldots,\emptyset}$ 就是答案。于是就有：

一道好题

$$f_{S_1,S_2,\dots,S_n}=\sum_{S_i\subseteq T_i} g_{T_1,T_2,\dots,T_n} \tag{5}$$

该式符合 (∗∗) 的左式，利用右式并代入 $S_i=\emptyset$ 则有：

$$g_{\emptyset,\emptyset,\dots,\emptyset}=\sum_{T_i}(-1)^{\sum_i |T_i|} f_{T_1,T_2,\dots,T_n}=\sum_{T_i}(-1)^{\sum_i |T_i|}\frac{\left(\sum_{i=1}^{n}(a_i-|T_i|)\right)!}{\prod_{i=1}^{n}(a_i-|T_i|)!} \tag{6}$$

显然，对于不同的 T_i，答案只与 $|T_i|$ 的值有关。也就是说，这里的集合中的元素实际上也有相同的地位（这一点在刚开始并不非常显然），于是我们可以模仿二项式反演的推导进行优化，改为枚举 $b_i=|T_i|$：

$$g_{\emptyset,\emptyset,\dots,\emptyset}=\sum_{0\le b_i\le a_i}(-1)^{\sum_i b_i}\left(\prod_{i=1}^{n}\binom{a_i-1}{b_i}\right)\frac{\left(\sum_{i=1}^{n}(a_i-b_i)\right)!}{\prod_{i=1}^{n}(a_i-b_i)!} \tag{7}$$

这个式子看上去已经大有可为了，我们在分治 FFT 的部分继续推导这个式子。

不等关系

给定一个字符串 $s_1, s_2, \ldots, s_n$，仅包含 $<$ 和 $>$ 两种字符。

你需要计算「使得 $p_i < p_{i+1}$ 当且仅当 s_i 为 $<$ 的排列 $p_1, p_2, \ldots, p_{n+1}$」的数量。

$n \le 10^5$

不等关系

同样，排列决定了计算难以从整体划分为局部。仍然考虑间接计数，首先寻找能够方便地整体计算的性质。

比较容易发现，同时存在 $<$ 的限制和 $>$ 的限制非常麻烦，不妨改为 $<$ 的限制和不做限制两种。我们假设集合 $S \subseteq \{1, 2, \ldots, n\}$ 中的位置有 $<$ 的限制，其余则不做限制。此时，排列 p 的关系由若干被 $<$ 连接的段组成，那么对于其中的任意一段，显然应该是升序的。只要分配给每一段的元素确定了，那么结果就唯一确定。这又是一个非常简单的组合问题：假设连接而成的段的长度分别是 $l_1, l_2, \ldots, l_m$，则方案数目为 $\frac{(n+1)!}{\prod_{i=1}^{m} l_i!}$。

第一步已经完成，现在考虑第二步：判断如何确定权值使得这些方案数的加权和为所求答案。考虑我们现在能够统计的是什么，就是满足 S 中的位置有 $<$ 的限制的排列方案数，将其记为 f_S，同样，将满足**恰好** S 中的位置有 $<$ 的限制的排列方案数记为 g_S，则有：

$$f_S = \sum_{S \subseteq T} g_T \tag{8}$$

不等关系

依然是利用 $(**)$ 的右式，假设题目中给出的需要满足 $<$ 的限制的位置集合是 S，那么所求的答案就是 g_S。于是：

$$g_S = \sum_{S \subseteq T} (-1)^{|T|-|S|} f_T = \sum_{S \subseteq T} (-1)^{|T|-|S|} \frac{(n+1)!}{\prod_{i=1}^{m} l_i!} \tag{9}$$

容易发现，由于对于不同的 T，得到的答案并不仅仅与 $|T|$ 有关，因此不能再次使用枚举 $|T|$ 的优化。我们将在动态规划的部分将这个式子转化为适合动态规划的形式，最后，我们在分治 FFT 的讲解中加速动态规划的过程来最终解决问题。

「CTS2019」氪金手游

题意简述：给定一棵 n 个点的树，每条边有一个方向。现在我们不断从 n 个点中随机抽取一个，直到所有的点都被抽取至少一次。每个点有一个取值为 $1,2,3$ 中的值的参数 p_i，代表这个点被抽中的概率为 $\frac{p_i}{\sum_j p_j}$。记 i 号点在第 T_i 次抽取时第一次被抽中，现在想要问你，满足“对于树上的任意一条有向边 $u_i \to v_i$，满足 $T_{u_i} < T_{v_i}$”的概率。

此外，p_i 的值不是给定的，它在一开始以 $P_{i,j}$ 的概率取值为 j，此后就不再变化。

$n \le 1000$

「CTS2019」氪金手游

题目所给的概率过于复杂，我们先暂时简化一下题目，假设 p_i 的值是确定的。

题目将限制关系给出了一棵树的形状，但是却没有统一的方向，这让人非常疑惑。我们不妨先进一步假设：给定的树边方向是有规律的，存在一个树根 r，使得边的方向都是父亲指向儿子。

那么，r 显然必须最先被抽中，这样的概率就是 $\frac{p_r}{\sum_j p_j}$。满足这个性质后，剩下的性质变为了在若干棵子树内部的限制。可以看出，不同子树之间没有限制关系，因此顺序任意，也就有了一定的"独立性"：即只要这些子树内部点的顺序分别被满足，那么总体就被满足。于是问题可以递归下去，每一步都类似 r 操作。

那么，我们假设 S_i 代表 i 的子树，答案就是：$\prod_i \frac{p_i}{\sum_{j\in S_i} p_j}$。

「CTS2019」氪金手游

不过，实际上树的方向不是那么统一的。我们仿照“不等关系”中的操作，将“父亲先于儿子”和“儿子先于父亲”两种限制改为“父亲先于儿子”和没有限制两种。假设有“父亲先于儿子”限制的树边集合是 S，于是，限制树现在变为了若干边方向统一的连通块。显然每个连通块可以独立计算答案，仍然假设 S_i 表示 i 所在连通块中 i 的子树，那么答案仍然为 $\prod_i \frac{p_i}{\sum_{j\in S_i} p_j}$。

第一步已经完成，现在考虑第二步：判断如何确定权值使得这些概率的加权和为所求答案。考虑我们现在能够统计的是什么，就是满足 S 中的位置有“父亲先于儿子”的限制的概率，将其记作 f_S，仍然仿照之前的做法，将满足**恰好** S 中的位置有“父亲先于儿子”的限制的概率记为 g_S。我们仍然有 $(**)$ 的左式成立。

「CTS2019」氪金手游

假设题中给定的需要满足“父亲先于儿子”限制的树边集合是 S，那么所求答案即为 g_S。利用 (**) 的右式，可得：

$$g_S=\sum_{S\subseteq T}(-1)^{|T|-|S|}f_T=\sum_{S\subseteq T}(-1)^{|T|-|S|}\prod_i\frac{p_i}{\sum_{j\in S_i}p_j} \tag{10}$$

现在，我们重新回到题目，关注 p_i 的取值并非开始就确定的问题。那么不妨枚举所有可能的取值，将它们的答案乘以概率相加：

$$g_S=\sum_{S\subseteq T}(-1)^{|T|-|S|}f_T=\sum_{S\subseteq T}\sum_{p_i}(-1)^{|T|-|S|}\left(\prod_i P_{i,p_i}\right)\prod_i\frac{p_i}{\sum_{j\in S_i}p_j} \tag{11}$$

现在，这个式子已经完全表示了答案。我们将在动态规划的部分讨论如何计算这个式子。

「CTS2019」随机立方体

简要题意：给定 $n \times m \times l$ 的立方体，向所有位置中填入均匀随机排列的 $1 \sim nml$ 的数字。我们称一个位置是极大的，如果这个位置的元素不小于于任意至少在 x, y, z 中有一维坐标相同的位置上的元素。求极大的位置恰好有 k 个的概率。

$n, m, l \leq 5 \cdot 10^5$

「CTS2019」随机立方体

本题几乎所有的题解开始的步骤如下：

“我们考虑到计算恰好 k 个极大的概率 p_k 非常不方便，不妨改为计算至少 k 个的概率 q_k。那么显然有：

$$q_i = \sum_{i \le j} \binom{j}{i} p_j$$

二项式反演，我们就得到了：

$$p_i = \sum_{i \le j} (-1)^{j-i} \binom{j}{i} q_j$$

”

「CTS2019」随机立方体

然而，这里有一些细节其实不清晰：

「CTS2019」随机立方体

然而，这里有一些细节其实不清晰：

- 不同的极大的位置是否具有良好的地位等价性，支持我们无视具体位置的不同直接对数目进行统计？

「CTS2019」随机立方体

然而，这里有一些细节其实不清晰：

- 不同的极大的位置是否具有良好的地位等价性，支持我们无视具体位置的不同直接对数目进行统计？
- “至少 k 个极大的概率”，是指极大位置的个数 $\geq k$ 的概率吗？如果是，关系式为什么不是 $q_i = \sum_{i\leq j} p_j$？

「CTS2019」随机立方体

然而，这里有一些细节其实不清晰：

- 不同的极大的位置是否具有良好的地位等价性，支持我们无视具体位置的不同直接对数目进行统计？
- “至少 k 个极大的概率”，是指极大位置的个数 $\geq k$ 的概率吗？如果是，关系式为什么不是 $q_i = \sum_{i\leq j} p_j$?
- 这个转化似乎不符合之前所述的两种二项式反演中的任何一种，这为什么是对的？

「CTS2019」随机立方体

然而，这里有一些细节其实不清晰：

- 不同的极大的位置是否具有良好的地位等价性，支持我们无视具体位置的不同直接对数目进行统计？
- “至少 k 个极大的概率”，是指极大位置的个数 $\geq k$ 的概率吗？如果是，关系式为什么不是 $q_i = \sum_{i\leq j} p_j$？
- 这个转化似乎不符合之前所述的两种二项式反演中的任何一种，这为什么是对的？

正是因为我们对处理的量的含义不够明确，只有模糊的认识就继续下一步，才导致我们常常觉得容斥捉摸不定，有时似乎很难掌握。让我们先通过另一种方式用清晰的思路解决问题，最后回来解答这个做法中的疑问。

「CTS2019」随机立方体

极大的概念是一个新概念，最好先观察一下它的性质。一个重要的观察是，如果两个位置都是极大的，那么它们的 x, y, z 坐标互不相同。

我们令全集 $U = \{(1,1,1),(1,1,2),\dots,(n,m,l)\}$ 为所有位置的集合。仍然和之前的题目一样，如果我们指定一个集合 $S \subseteq U$，指定极大位置**恰好**是 S 中的元素，那么我们实际上指定了两种限制：是极大位置和不是极大位置，对象分别是 S 和 $U \setminus S$，比较难以统计。如果我们将限制修改为“是极大位置”和“不做限制”两种，似乎就变得可行了很多。也就是说，现在指定一个集合 $S \subseteq U$，令 S 中的位置是极大位置，其余位置随意，想要统计这样的情况发生的概率，记为 f_S。

此时我们就需要利用刚刚观察到的性质：如果两个位置都是极大的，那么它们的 x, y, z 坐标互不相同。于是，如果 S 中有两个元素有一维坐标相同，那么 $f_S = 0$。现在，只需要考虑 S 中的位置满足任意一维坐标两两不同的情况。

「CTS2019」随机立方体

任意一维坐标两两不同，就意味着任何一个点的任何一个坐标都是独一无二的（这暗示了 $|S| \le \min\{n, m, l\}$ 时，f_S 才可能不为 0）。此时，极大值的位置才体现出良好的地位等价性：极大值的判定只与坐标是否相同有关，而与具体的坐标值无关，因此对每一维坐标做置换，并不会影响是否为极大值的结果。那么，不妨通过置换坐标，将 S 中的点置换到 $(1,1,1),(2,2,2),\dots,(|S|,|S|,|S|)$ 上，方便我们进行分析。

现在只需要求解 $(1,1,1),(2,2,2),\dots,(|S|,|S|,|S|)$ 是极大位置的概率。

「CTS2019」随机立方体

任意一维坐标两两不同，就意味着任何一个点的任何一个坐标都是独一无二的（这暗示了 $|S| \leq \min\{n, m, l\}$ 时，f_S 才可能不为 0）。此时，极大值的位置才体现出良好的地位等价性：极大值的判定只与坐标是否相同有关，而与具体的坐标值无关，因此对每一维坐标做置换，并不会影响是否为极大值的结果。那么，不妨通过置换坐标，将 S 中的点置换到 $(1,1,1),(2,2,2),\ldots,(|S|,|S|,|S|)$ 上，方便我们进行分析。

现在只需要求解 $(1,1,1),(2,2,2),\ldots,(|S|,|S|,|S|)$ 是极大位置的概率。

思考一下，是否可以这么考虑：对于特定的 (i,i,i)，如果它是极大的，那么它就是和它至少一维坐标相同的 $nm+nl+ml-n-m-l+1$ 个点中的最大值，由于每个位置的值取法相同，因此它是最大值的概率就是 $\frac{1}{nm+nl+ml-n-m-l+1}$，从而 $(1,1,1),(2,2,2),\ldots,(|S|,|S|,|S|)$ 是极大位置的概率就是 $\left(\frac{1}{nm+nl+ml-n-m-l+1}\right)^{|S|}$？

「CTS2019」随机立方体

这显然是不对的，一个简短的解释是“(i,i,i) 是极大位置”这一事件对于不同的 i 并非独立事件，因此概率不能直接相乘。如果说的更具体一些，在我们假定“$(1,1,1)$ 是极大位置”之后，$(1,2,2)$ 的地位和 $(3,2,2)$ 显然就不同了，因为前者已经被假定小于某个位置的值，而后者尚没有任何假设，那么“每个位置的值取法相同”这个前提也就是伪的。

是否有方法能够使我们假定某个位置是极大值之后，剩下的位置地位仍然相同呢？

那么再次考虑：如果假定“$(1,1,1)$ 是极大位置”，坐标至少有一维为 1 的位置肯定需要被假设小于 $(1,1,1)$ 的值，如果其它位置没有这个假设，那么彼此地位就不等价。不过，如果假设 $(1,1,1)$ 是所有至少有一维坐标 $\le |S|$ 的位置中最大的，那么所有其它位置都被加上了小于 $(1,1,1)$ 的假设，从而地位仍然相等。

可以发现，如果要这样假设，那么 $(1,1,1)$ 必须是极大位置中值最大的。这提示我们给所有极大位置加上序关系的假设：由于这些位置地位相同，因此任意一种序关系出现的概率相等，只要考虑其中一种，最后给概率乘以 $|S|!$ 即可。

「CTS2019」随机立方体

那么不妨采取特殊的序关系 $v_{1,1,1} < v_{2,2,2} < \cdots < v_{|S|,|S|,|S|}$。在这样的假设下，“$(i, i, i)$ 是极大位置”的条件可以被修改为“(i, i, i) 的值是所有至少有一维 $\leq i$ 的位置中最大的”。确定序关系的关键在于，现在被限制的位置的范围是互相包含的。那么，我们从大到小考虑，考虑“(i, i, i) 的值是所有至少有一维 $\leq i$ 的位置中最大的”的条件时，这些位置的地位仍然等价，于是概率就是 $\frac{1}{a_i}$，其中 a_i 表示所有至少有一维 $\leq i$ 的位置的数目。这样以后，对于所有至少有一维 $\leq i-1$ 的位置，由于它们都被添加了“小于 (i, i, i)”的假设，因此地位仍然相等。

重新考虑序关系的影响，通过以上步骤我们终于得到：

$$f_S = |S|! \prod_{i=1}^{|S|} \frac{1}{a_i} \tag{a}$$

「CTS2019」随机立方体

第一步已经完成，现在考虑第二步：判断如何确定权值使得这些概率的加权和为所求答案。考虑我们现在能够统计的是什么，就是满足 S 中的位置是极大位置的概率 f_S，仍然仿照之前的做法，将满足**恰好** S 中的位置是极大值的概率记为 g_S。我们仍然有 $(**)$ 的左式成立。于是利用 $(**)$ 的右式，就得到：

$$g_S = \sum_{S \subseteq T} (-1)^{|T|-|S|} f_T \tag{12}$$

这里需要注意一点，那就是 f_S 的计算并不完全按照 (a) 式进行，当 S 中有两个位置有某一维坐标相同时，$f_S = 0$，因此，我们不能直接利用 (a) 式的结果只与 $|S|$ 有关，简化这个表达式。那么，如果 S 中有两个位置有某一维坐标相同，显然作为 S 的超集 T 也有同样的问题，于是这种情况下 $g_S = 0$。

否则，S 中任意一维坐标两两不同。我们想要利用 (a) 式优化求和，那么：指定 $|T|$，如果希望 T 仍然满足任意一维坐标两两不同，我们需要在 S 已经选定的坐标之外取值。

「CTS2019」随机立方体

这是一个简单的组合问题。先使得 $T\setminus S$ 的元素暂时变得有序，那么对第一维来说，选择的方案数显然是
$(n-|S|)(n-|S|-1)\cdots(n-|T|+1)=(n-|S|)^{\underline{|T|-|S|}}$，另外两维类似。最后除以顺序 $(|T|-|S|)!$，于是对特定的 $|T|$，能够产生贡献的 T 的数目是：

$$\frac{(n-|S|)^{\underline{|T|-|S|}}(m-|S|)^{\underline{|T|-|S|}}(l-|S|)^{\underline{|T|-|S|}}}{(|T|-|S|)!} \tag{13}$$

于是改为枚举 $|T|$，求解 g_S 的步骤可以优化为：

$$g_S=\sum_{i=|S|}^{\min\{n,m,l\}}(-1)^{i-|S|}\frac{(n-|S|)^{\underline{i-|S|}}(m-|S|)^{\underline{i-|S|}}(l-|S|)^{\underline{i-|S|}}}{(i-|S|)!}i!\prod_{j=1}^{i}\frac{1}{a_j} \tag{14}$$

注意，这个式子仅在 S 中任意一维坐标两两不同的情况下成立。

「CTS2019」随机立方体

题目要求的是恰好有 k 个极大位置，但不要求具体位置，于是我们还需要枚举所有能够产生贡献的 S，类似于刚刚的做法，方案数是：

$$\frac{n^{\underline{k}}m^{\underline{k}}l^{\underline{k}}}{k!} \tag{15}$$

将这个数字与之前的式子相乘就是答案：

$$\frac{n^{\underline{k}}m^{\underline{k}}l^{\underline{k}}}{k!}\sum_{i=k}^{\min\{n,m,l\}}(-1)^{i-k}\frac{(n-k)^{\underline{i-k}}(m-k)^{\underline{i-k}}(l-k)^{\underline{i-k}}}{(i-k)!}i!\prod_{j=1}^{i}\frac{1}{a_j} \tag{16}$$

整理得到答案的形式：

$$\sum_{i=k}^{\min\{n,m,l\}}(-1)^{i-k}n^{\underline{i}}m^{\underline{i}}l^{\underline{i}}\binom{i}{k}\prod_{j=1}^{i}\frac{1}{a_j} \tag{17}$$

如果利用线性求逆元，算法复杂度就是 $O(n)$。

「CTS2019」随机立方体

现在回过头来看一般题解中存在的问题。首先，在没有排除坐标相同的状况之前，极大位置的地位等价性并不非常显然。因此，实际做题时，最好先不要盲目利用地位等价性化简推理过程，不妨先直接枚举子集并利用容斥原理而非二项式反演，这样思路会更加清晰。并且，如果满足一定的对称性，在后续的式子中也可以进行化简。

再看看关于“至少 k 个极大位置”的定义问题。其实，这是一个很有误导性的说法，并且广泛存在在各类容斥题目中。此处“至少”说法是错误的，q_k 的准确定义应当为对所有“至少有某特定 k 个位置为极大位置的概率”的代数和。因此，q_k 并非一个 $[0,1]$ 之间的数字，这会在解题过程中造成一定的干扰。

最后，那个最初出现的转化式需要一定的推导，并非二项式反演的典型结论。

总的来说，虽然以主流的题解形式解题看上去过程更简单，但其实我们混淆和忽略了许多内容。如果仅满足于用这样的方式解题，很可能在下一次遇见类似问题时仍然觉得无从下手。从最基本的容斥原理出发，仔细讨论直接推理，是一种更好的方式。

曲线救国

简化题意：对于所有 n 个点的简单无向图 G，求所有“连通块中的点编号的最小值”的最大值为 k 的图的数目，对 $k=1,2,\ldots,n$ 求答案。$n \le 5 \cdot 10^5$

曲线救国

说到最小值的最大值，就想到二分答案。这是因为想要判断这个值是否 $\le t$ 是容易的：只要每个连通块编号最小值都 $\le t$ 即可。那么此处显然可以用一个差分简化问题：统计 a_t 代表所有“连通块中的点编号的最小值”的最大值 $\le t$ 的方案数，那么 $a_k - a_{k-1}$ 就是 k 的答案。于是只要关心如何求出 $a_1, a_2, \ldots, a_n$。（利用差分把等于的限制转化为小于等于的限制，也是一种常用的间接计数法）

要求 a_t，就是要求所有连通块中都要包含至少一个编号 $\le t$ 的点。但“至少一个”看起来很麻烦，我们更喜欢和它恰好相反的“不存在”，也就是说，如果将限制改为强制某些特定的连通块中不存在 $\le t$ 的点，会不会更容易进行整体统计？

一个可能的想法是，枚举 $> t$ 的点中，单独形成连通块的点的点集，令其为 S。不过，实际情况下可能有 S 以外的点也形成了最小编号 $> t$ 的连通块，并且还能和 S 中的元素在同一连通块中。这样，统计变得很麻烦，难以整体计算。

曲线救国

之前做法的困难之处在于，连通块中的元素不是确定的。因此，我们不妨干脆把哪些点在同一连通块中也确定下来。也就是说，枚举集合的集合 $\{S_1, S_2, \ldots, S_k\}(S_i \subseteq \{t+1, t+2, \ldots, n\})$，使得 S_i 中的点组成一个连通块，其余的点则不做限制。注意到，存在合法方案有一个必要条件：对于任意 $1 \le i < j \le k$，S_i 与 S_j 不相交。这种情况下的方案数是很好统计的：首先考虑剩下的 $n - \sum_i |S_i|$ 个点，随意组成连通块，有 $g_{n-\sum_i |S_i|}$ 种方法，其中 $g_n = 2^{\binom{n}{2}}$ 是 n 个带标号点组成的无向图数。其次，既然已经确定了每个连通块的点集合，那么对每个连通块分别统计方案数再相乘即可。答案为 $\prod_i f_{|S_i|}$，其中 f_n 是 n 个带标号点组成的连通无向图数。

题外话：正确看待 exp 在带标号计数中的组合意义

各位应该已经了解了 exp 在指数生成函数的计数中代表对对象进行背包之后的结果，而 ln 则代表其反演。但这一方法的证明在很多题解中讲述得并不清楚，因此在这里再讲一遍。

现在有一个组合结构 T，每个 T 中的元素 t 都有一个大小 $|t|$，表示 t 是由编号 $1, 2, \dots, |t|$ 的点通过一些规则组合而成的结构。有一个描述组合结构 T 的生成函数 f，其中 $[x^n]f$ 代表 T 中大小为 n 的元素的个数。现在，我们希望统计这样的元素 t' 的个数：它的大小为 $|t'| = n$，并且由若干个结构与 T 中元素几乎相同的元素 t 组合而成，唯一的不同在于 t 被分配的标号并非 $1, 2, \dots, |t|$，而是分别在 t' 的 $1, 2, \dots, |t'|$ 这些编号中选出了独一无二的 $|t|$ 个，以此分配了 t' 的标号。

为了统计这一个数，首先观察到在确定每个组成 t' 的 t 编号独一无二后，它们必然是彼此不同的，因此可以先使得它们变得有序。我们枚举 t 的数目 k，接下来，首先依次确定 t 的大小 $a_1, a_2, \dots, a_k$，然后决定怎样分配标号。然后根据独立性分别取每一个 t 可能的结构数目再相乘，最后除以 $k!$ 消除顺序。

题外话：正确看待 exp 在带标号计数中的组合意义

将上述过程表述为式子如下：

$$[x^n]g=\sum_{k\geq 0}\frac{1}{k!}\sum_{a_1+a_2+\cdots+a_k=n}\binom{n}{a_1,a_2,\ldots,a_k}\prod_{i=1}^{k}[x^{a_i}]f \tag{18}$$

其中 g 代表 T' 的生成函数。将组合数用阶乘分解并重新安排顺序得到：

$$\frac{[x^n]g}{n!}=\sum_{k\geq 0}\frac{1}{k!}\sum_{a_1+a_2+\cdots+a_k=n}\prod_{i=1}^{k}\frac{[x^{a_i}]f}{a_i!} \tag{19}$$

注意到 $\sum_{a_1+a_2+\cdots+a_k=n}\prod_{i=1}^{k}\frac{[x^{a_i}]f}{a_i!}$ 正是 $[x^n]\tilde{f}^k$，其中 $\tilde{f}$ 代表 f 的指数生成函数，于是：

$$[x^n]\tilde{g}=[x^n]\sum_{k\geq 0}\frac{\tilde{f}^k}{k!}=[x^n]e^{\tilde{f}} \tag{20}$$

这样就证明了结论。

曲线救国

继续考虑刚刚的问题，由 $\tilde{g} = e^{\tilde{f}}$，可知 $f_n = n![x^n]\ln\tilde{g}$，可以在 $O(n\log n)$ 的时间内整体计算。于是我们已经得到包含由 $S_1, S_2, \ldots, S_k$ 这些 $> t$ 的点组成连通块的方案数，记为 $F_{\{S_1,S_2,\ldots,S_k\}}$：

$$F_{\{S_1,S_2,\ldots,S_k\}} = g_{n-\sum_i |S_i|}\prod_{i=1}^{k} f_{|S_i|} \tag{21}$$

这样，我们依然可以仿照前几题的做法，假设 $G_{\{S_1,S_2,\ldots,S_k\}}$ 表示恰好包含由 $S_1, S_2, \ldots, S_k$ 这些 $> t$ 的点组成连通块的方案数。令全集 U 为所有 $\{k, k+1, \ldots, n\}$ 的子集组成的集合，就有：

$$F_{\{S_1,S_2,\ldots,S_k\}} = \sum_{\{S_1,S_2,\ldots,S_k\}\subseteq\{T_i\}\subseteq U} G_{\{T_i\}} \tag{22}$$

曲线救国

仍然利用 (**) 的右式，得到：

$$G_{\{S_1,S_2,\ldots,S_k\}} = \sum_{\{S_1,S_2,\ldots,S_k\}\subseteq\{T_i\}\subseteq U} (-1)^{|\{T_i\}|-k} F_{\{T_i\}} \tag{23}$$

我们要求不含有任何由 $\geq t$ 的点组成的连通块，于是就是要求 $G_{\emptyset}$：

$$G_{\emptyset} = \sum_{\{T_i\}\subseteq U} (-1)^{|\{T_i\}|} F_{\{T_i\}} \tag{24}$$

考虑如何计算该式。首先，需要保证 $\{T_i\}$ 中的元素两两不相交。保证这一点以后，答案只和每个元素的大小有关（这一点和随机立方体很像，存在某个性质，不满足这个性质的位置答案为 0，否则计算规则是相似的）。由于 T_i 在保证不相交的情况下互不相同，我们仍然可以仿照随机立方体的做法以及 exp 的推导过程，将其暂时转化为有序的。

曲线救国

那么首先枚举连通块个数 k，预除以 $k!$ 即可消除有序的影响。接下来枚举 $a_i = |T_i|$，用组合数分配其标号，最后乘以连通块具体形态方案数 $F_{\{T_i\}}$ 即可：

$$G_\emptyset = \sum_k \frac{(-1)^k}{k!} \sum_{\sum_i a_i \le n-t} \left(\prod_i \binom{n-t-\sum_{j=1}^{i-1} a_j}{a_i} \right) g_{n-\sum_i a_i} \prod_{i=1}^{k} f_{a_i} \quad (25)$$

进行几步推导将组合数转化为阶乘并分配：

$$G_\emptyset = \sum_k \frac{(-1)^k}{k!} \sum_{\sum_i a_i \le n-t} \binom{n-t}{a_1, a_2, \ldots, n-t-\sum_i a_i} g_{n-\sum_i a_i} \prod_{i=1}^{k} f_{a_i} \quad (26)$$

$$G_\emptyset = (n-t)! \sum_k \frac{(-1)^k}{k!} \sum_{\sum_i a_i \le n-t} \frac{g_{n-\sum_i a_i}}{(n-t-\sum_i a_i)!} \prod_{i=1}^{k} \frac{f_{a_i}}{a_i!} \quad (27)$$

曲线救国

$$G_\emptyset=(n-t)!\sum_k\frac{(-1)^k}{k!}\sum_{\sum_i a_i\le n-t}\frac{g_{n-\sum_i a_i}}{(n-t-\sum_i a_i)!}\prod_{i=1}^k\frac{f_{a_i}}{a_i!}\tag{28}$$

可以发现，g 的次数和阶乘的数量不太匹配，将 $\sum_{\sum_i a_i\le n-t}\frac{g_{n-\sum_i a_i}}{(n-t-\sum_i a_i)!}\prod_{i=1}^k\frac{f_{a_i}}{a_i!}$ 转化为某个多项式的某一项系数比较困难。那么不妨退而求其次，通过枚举 $i=\sum_j a_j$ 的值，将 $\sum_{\sum_j a_j=i}\prod_{j=1}^k\frac{f_{a_j}}{a_j!}$ 转化为多项式的系数：

$$G_\emptyset=(n-t)!\sum_k\frac{(-1)^k}{k!}\sum_{i=0}^{n-t}\frac{g_{n-i}}{(n-t-i)!}\sum_{\sum_j a_j=i}\prod_{j=1}^k\frac{f_{a_j}}{a_j!}\tag{29}$$

曲线救国

从而显然有：

$$G_{\emptyset}=(n-t)!\sum_k\frac{(-1)^k}{k!}\sum_{i=0}^{n-t}\frac{g_{n-i}}{(n-t-i)!}[x^i]\tilde{f}^k \tag{30}$$

交换求和号，并转化：

$$G_{\emptyset}=(n-t)!\sum_{i=0}^{n-t}\frac{g_{n-i}}{(n-t-i)!}[x^i]\sum_k\frac{(-\tilde{f})^k}{k!} \tag{31}$$

显然，$\sum_k\frac{(-\tilde{f})^k}{k!}$ 可以被转化成 $e^{-\tilde{f}}$，这里有一个小技巧：由于 $\tilde{f}=\ln\tilde{g}$，因此实际上 $e^{-\tilde{f}}=e^{-\ln\tilde{g}}=\frac{1}{\tilde{g}}$，故只需要多项式求逆就可以求出 $e^{-\tilde{f}}$，这大大减小了常数。

曲线救国

最后对式子进行一些整理：

$$G_\emptyset = (n-t)!\sum_{i=0}^{n-t}\frac{g_{n-i}}{(n-t-i)!}[x^i]\frac{1}{\tilde{g}} = (n-t)!\sum_{i=0}^{n-t}\frac{1}{(n-t-i)!}\left(g_{n-i}[x^i]\frac{1}{\tilde{g}}\right) \tag{32}$$

最后的和式明显地变为了一个卷积式，通过 FFT 可以 $O(n\log n)$ 求出每一项。于是就在 $O(n\log n)$ 内解决了问题。

我们已经对五道质量较高的容斥题目的容斥部分进行了分析。现在，让我们对这些题目的容斥方法进行对比和总结，找到它们应用容斥方法的共通点。

容斥的对象

这五道题均用了容斥原理的 (**) 式，实际上 (**) 式确实更常用一些。不过 (*) 和 (**) 其实是可以等价转化的，因此对两种方式都要熟悉。

我们就以考虑 (**) 式

$f_S = \sum_{S\subseteq T\subseteq U} g_T \Longleftrightarrow g_S = \sum_{S\subseteq T\subseteq U}(-1)^{|T|-|S|} f_T$ 为例。要让这个式子应用到题目中，我们就必须指明 S, T, U 是什么，而 f_S 和 g_S 的含义又是什么。

容斥的对象

这五道题均用了容斥原理的 (**) 式，实际上 (**) 式确实更常用一些。不过 (*) 和 (**) 其实是可以等价转化的，因此对两种方式都要熟悉。

我们就以考虑 (**) 式 $f_S=\sum_{S\subseteq T\subseteq U} g_T \iff g_S=\sum_{S\subseteq T\subseteq U}(-1)^{|T|-|S|}f_T$ 为例。要让这个式子应用到题目中，我们就必须指明 S, T, U 是什么，而 f_S 和 g_S 的含义又是什么。

- 一道好题：U 表示所有“序列中的第 j 个颜色为 i 的球与序列中的第 $j+1$ 个颜色为 i 的球”这些二元球对的集合。f_S 代表集合 $S\subseteq U$ 中的球对相邻的方案数，g_S 代表**恰好**集合 $S\subseteq U$ 中的球对相邻的方案数。

容斥的对象

这五道题均用了容斥原理的 (**) 式，实际上 (**) 式确实更常用一些。不过 (*) 和 (**) 其实是可以等价转化的，因此对两种方式都要熟悉。

我们就以考虑 (**) 式 $f_S = \sum_{S\subseteq T\subseteq U} g_T \iff g_S = \sum_{S\subseteq T\subseteq U}(-1)^{|T|-|S|} f_T$ 为例。要让这个式子应用到题目中，我们就必须指明 S, T, U 是什么，而 f_S 和 g_S 的含义又是什么。

- 一道好题：U 表示所有“序列中的第 j 个颜色为 i 的球与序列中的第 $j+1$ 个颜色为 i 的球”这些二元球对的集合。f_S 代表集合 $S \subseteq U$ 中的球对相邻的方案数，g_S 代表**恰好**集合 $S \subseteq U$ 中的球对相邻的方案数。
- 不等关系：U 表示所有“序列中的第 i 个元素和第 $i+1$ 个元素”这些二元对的集合。f_S 代表集合 $S \subseteq U$ 中的二元对满足前者小于后者的方案数，g_S 代表**恰好**集合 $S \subseteq U$ 中的二元对满足前者小于后者的方案数。

容斥的对象

这五道题均用了容斥原理的 (**) 式，实际上 (**) 式确实更常用一些。不过 (*) 和 (**) 其实是可以等价转化的，因此对两种方式都要熟悉。

我们就以考虑 (**) 式

$f_S=\sum_{S\subseteq T\subseteq U}g_T \iff g_S=\sum_{S\subseteq T\subseteq U}(-1)^{|T|-|S|}f_T$ 为例。要让这个式子应用到题目中，我们就必须指明 S,T,U 是什么，而 f_S 和 g_S 的含义又是什么。

- 一道好题：U 表示所有“序列中的第 j 个颜色为 i 的球与序列中的第 $j+1$ 个颜色为 i 的球”这些二元球对的集合。f_S 代表集合 $S\subseteq U$ 中的球对相邻的方案数，g_S 代表**恰好**集合 $S\subseteq U$ 中的球对相邻的方案数。
- 不等关系：U 表示所有“序列中的第 i 个元素和第 $i+1$ 个元素”这些二元对的集合。f_S 代表集合 $S\subseteq U$ 中的二元对满足前者小于后者的方案数，g_S 代表**恰好**集合 $S\subseteq U$ 中的二元对满足前者小于后者的方案数。
- 氪金手游：U 表示所有在树上有父子关系的点对 (x,y)（x 是父亲）的集合。f_S 代表满足集合 $S\subseteq U$ 中的点对 (x,y) 均有 $T_x<T_y$ 的概率，g_S 代表满足**恰好**集合 $S\subseteq U$ 中的点对 (x,y) 均有 $T_x<T_y$ 的概率。

容斥的对象

这五道题均用了容斥原理的 (**) 式，实际上 (**) 式确实更常用一些。不过 (*) 和 (**) 其实是可以等价转化的，因此对两种方式都要熟悉。

我们就以考虑 (**) 式

$f_S=\sum_{S\subseteq T\subseteq U} g_T \iff g_S=\sum_{S\subseteq T\subseteq U}(-1)^{|T|-|S|}f_T$ 为例。要让这个式子应用到题目中，我们就必须指明 S,T,U 是什么，而 f_S 和 g_S 的含义又是什么。

- 一道好题：U 表示所有“序列中的第 j 个颜色为 i 的球与序列中的第 $j+1$ 个颜色为 i 的球”这些二元球对的集合。f_S 代表集合 $S\subseteq U$ 中的球对相邻的方案数，g_S 代表**恰好**集合 $S\subseteq U$ 中的球对相邻的方案数。
- 不等关系：U 表示所有“序列中的第 i 个元素和第 $i+1$ 个元素”这些二元对的集合。f_S 代表集合 $S\subseteq U$ 中的二元对满足前者小于后者的方案数，g_S 代表**恰好**集合 $S\subseteq U$ 中的二元对满足前者小于后者的方案数。
- 氪金手游：U 表示所有在树上有父子关系的点对 (x,y)（x 是父亲）的集合。f_S 代表满足集合 $S\subseteq U$ 中的点对 (x,y) 均有 $T_x<T_y$ 的概率，g_S 代表满足**恰好**集合 $S\subseteq U$ 中的点对 (x,y) 均有 $T_x<T_y$ 的概率。
- 随机立方体：U 表示立方体中所有位置的集合。f_S 代表集合 $S\subseteq U$ 中的位置是极大位置的概率，g_S 代表**恰好**集合 $S\subseteq U$ 中的位置是极大位置的概率。

容斥的对象

这五道题均用了容斥原理的 (**) 式，实际上 (**) 式确实更常用一些。不过 (*) 和 (**) 其实是可以等价转化的，因此对两种方式都要熟悉。

我们就以考虑 (**) 式

$f_S=\sum_{S\subseteq T\subseteq U} g_T \iff g_S=\sum_{S\subseteq T\subseteq U}(-1)^{|T|-|S|}f_T$ 为例。要让这个式子应用到题目中，我们就必须指明 S,T,U 是什么，而 f_S 和 g_S 的含义又是什么。

- 一道好题：U 表示所有“序列中的第 j 个颜色为 i 的球与序列中的第 $j+1$ 个颜色为 i 的球”这些二元球对的集合。f_S 代表集合 $S\subseteq U$ 中的球对相邻的方案数，g_S 代表**恰好**集合 $S\subseteq U$ 中的球对相邻的方案数。
- 不等关系：U 表示所有“序列中的第 i 个元素和第 $i+1$ 个元素”这些二元对的集合。f_S 代表集合 $S\subseteq U$ 中的二元对满足前者小于后者的方案数，g_S 代表**恰好**集合 $S\subseteq U$ 中的二元对满足前者小于后者的方案数。
- 氪金手游：U 表示所有在树上有父子关系的点对 (x,y)（x 是父亲）的集合。f_S 代表满足集合 $S\subseteq U$ 中的点对 (x,y) 均有 $T_x<T_y$ 的概率，g_S 代表满足**恰好**集合 $S\subseteq U$ 中的点对 (x,y) 均有 $T_x<T_y$ 的概率。
- 随机立方体：U 表示立方体中所有位置的集合。f_S 代表集合 $S\subseteq U$ 中的位置是极大位置的概率，g_S 代表**恰好**集合 $S\subseteq U$ 中的位置是极大位置的概率。
- 曲线救国：U 表示所有 $>t$ 的点组成的点集的非空子集的集合。f_S 代表满足集合 $S\subseteq U$ 中的子集分别组成一个 $>t$ 的连通块的方案数，g_S 代表满足**恰好**集合 $S\subseteq U$ 中的子集分别组成一个 $>t$ 的连通块的方案数。

容斥的对象

从这些例子中我们可以看出，容斥对象的形式是多样的。它可以像随机立方体一样是单纯的位置的集合，也可以是一些点组成的二元组的集合，甚至可以像曲线救国一样是子集的集合……实际做题中，我们要根据题目的性质具体确定容斥对象。

确定集合 U 后，几乎可以立即确定 f_S 和 g_S 的含义。具体来说，我们需要额外引入一个性质 p 针对 U 中的元素。而 f_S 表示 S 中的元素满足性质 p 的概率/方案数，而 g_S 则是**恰好** S 中的元素满足性质 p 的概率/方案数（言外之意是，$U\setminus S$ 中的元素不满足性质 p)。通常，题目所求的是 g_S，而 f_S 因为限制宽松会更便于计算，于是利用 $(**)$ 的右式就可以用 f_S 推导出 g_S。

容斥导出的答案

再分别看看这些题目分别要求哪些 g_S:

容斥导出的答案

再分别看看这些题目分别要求哪些 g_S：

- 一道好题：求解任意同色对不相邻的方案数，是 $g_\emptyset$。

容斥导出的答案

再分别看看这些题目分别要求哪些 g_S：

- 一道好题：求解任意同色对不相邻的方案数，是 $g_{\emptyset}$。
- 不等关系：给定了取小于号的对 S 和取大于号的对 $U\setminus S$，于是答案是 g_S。

容斥导出的答案

再分别看看这些题目分别要求哪些 g_S：

- 一道好题：求解任意同色对不相邻的方案数，是 $g_\emptyset$。
- 不等关系：给定了取小于号的对 S 和取大于号的对 $U \setminus S$，于是答案是 g_S。
- 氪金手游：给定了取父亲早于儿子的对 S 和晚于儿子的对 $U \setminus S$，于是答案是 g_S。

容斥导出的答案

再分别看看这些题目分别要求哪些 g_S:

- 一道好题:求解任意同色对不相邻的方案数,是 $g_\emptyset$。
- 不等关系:给定了取小于号的对 S 和取大于号的对 $U \setminus S$,于是答案是 g_S。
- 氪金手游:给定了取父亲早于儿子的对 S 和晚于儿子的对 $U \setminus S$,于是答案是 g_S。
- 随机立方体:比较特殊,需要恰好 k 个极大位置,于是答案是 $\sum_{|S|=k} g_S$。

容斥导出的答案

再分别看看这些题目分别要求哪些 g_S：

- 一道好题：求解任意同色对不相邻的方案数，是 $g_\emptyset$。
- 不等关系：给定了取小于号的对 S 和取大于号的对 $U \setminus S$，于是答案是 g_S。
- 氪金手游：给定了取父亲早于儿子的对 S 和晚于儿子的对 $U \setminus S$，于是答案是 g_S。
- 随机立方体：比较特殊，需要恰好 k 个极大位置，于是答案是 $\sum_{|S|=k} g_S$。
- 曲线救国：所有 $> t$ 的点不能单独组成连通块，于是答案是 $G_\emptyset$。

容斥导出的答案

可见，求解 $g_\emptyset$ 和 g_S 都是常见的形式。求解 $g_\emptyset$ 的题目中，虽然看似只是要求所有元素都不满足性质 p，应该不会比求解 S 中元素满足性质 p 的 f_S 难求解。但实际上，有时“不满足性质 p”的情况比“满足性质 p”的情况复杂很多，因此还是以不满足性质 p 进行计数更合理。如“一道好题”中，同色的相邻元素可以合并，显然比强制同色不相邻好求解得多。

题目也可能让我们求解 g_S，这时，f_S 的优势往往在于不会对 $U \setminus S$ 中的元素有要求。当然，也可能“不满足性质 p”仍然更便于计数，此时我们需要使用 $(*)$ 式。

有些情形下，题目所求的答案会给定一个性质 p'，要求 $\sum_{p'(S)} g_S$，例如随机立方体对所有 $|S| = k$ 的 g_S 求和。此时，我们要在推导式子时灵活根据 p' 的性质对和式进行化简。例如，考虑到对 $|S| = k$ 的 g_S 求和，就可以查看 g_S 是否在 $|S|$ 相同时有着相似的结果。

化简容斥原理推导出的式子

鉴于容斥原理本身的性质，如果不对得到的和式进行任何化简，那么求解的复杂度一般至少是 $O(2^{|U|})$ 的。显然，出题人一般不喜欢将这样没有技术含量的做法作为标算，而是更喜欢利用题目的性质化简和式得到复杂度更低的做法。

我们看看之前所做的五道题分别是如何降低复杂度的（可以看出，作为优秀的容斥原理题，它们都没有采用直接计算的方式作为标算）：

化简容斥原理推导出的式子

鉴于容斥原理本身的性质，如果不对得到的和式进行任何化简，那么求解的复杂度一般至少是 $O(2^{|U|})$ 的。显然，出题人一般不喜欢将这样没有技术含量的做法作为标算，而是更喜欢利用题目的性质化简和式得到复杂度更低的做法。

我们看看之前所做的五道题分别是如何降低复杂度的（可以看出，作为优秀的容斥原理题，它们都没有采用直接计算的方式作为标算）：

- 一道好题：首先观察得到 $f_{\{S_i\}}$ 的取值仅和 $\{|S_i|\}$ 有关，于是改为枚举 $|S_i|$，用组合数计算满足这个条件的 S_i 的数目。进一步的化简将在分治 FFT 的部分呈现。

化简容斥原理推导出的式子

鉴于容斥原理本身的性质，如果不对得到的和式进行任何化简，那么求解的复杂度一般至少是 $O(2^{|U|})$ 的。显然，出题人一般不喜欢将这样没有技术含量的做法作为标算，而是更喜欢利用题目的性质化简和式得到复杂度更低的做法。

我们看看之前所做的五道题分别是如何降低复杂度的（可以看出，作为优秀的容斥原理题，它们都没有采用直接计算的方式作为标算）：

- 一道好题：首先观察得到 $f_{\{S_i\}}$ 的取值仅和 $\{|S_i|\}$ 有关，于是改为枚举 $|S_i|$，用组合数计算满足这个条件的 S_i 的数目。进一步的化简将在分治 FFT 的部分呈现。
- 不等关系：对于 f_S，由 S 导出了连续段的长度 $l_1, l_2, \ldots, l_m$，答案仅与 $\frac{1}{\prod_i l_i!}$ 有关。然而，对于不同的 S，导出的 $\{l_i\}$ 仍是不同的。具体的计算方式将在动态规划的部分呈现，并利用分治 FFT 进行加速。

化简容斥原理推导出的式子

鉴于容斥原理本身的性质，如果不对得到的和式进行任何化简，那么求解的复杂度一般至少是 $O(2^{|U|})$ 的。显然，出题人一般不喜欢将这样没有技术含量的做法作为标算，而是更喜欢利用题目的性质化简和式得到复杂度更低的做法。

我们看看之前所做的五道题分别是如何降低复杂度的（可以看出，作为优秀的容斥原理题，它们都没有采用直接计算的方式作为标算）：

- 一道好题：首先观察得到 $f_{\{S_i\}}$ 的取值仅和 $\{|S_i|\}$ 有关，于是改为枚举 $|S_i|$，用组合数计算满足这个条件的 S_i 的数目。进一步的化简将在分治 FFT 的部分呈现。
- 不等关系：对于 f_S，由 S 导出了连续段的长度 $l_1, l_2, \ldots, l_m$，答案仅与 $\frac{1}{\prod_i l_i!}$ 有关。然而，对于不同的 S，导出的 $\{l_i\}$ 仍是不同的。具体的计算方式将在动态规划的部分呈现，并利用分治 FFT 进行加速。
- 氪金手游：对于 f_S，由 S 导出了连通块上的子树 $S_1, S_2, \ldots, S_n$，答案仅与 $\prod_i \frac{p_i}{\sum_{j \in S_i} p_j}$ 有关。然而，对于不同的 S，导出的 $\{S_i\}$ 仍是不同的。具体的计算方式将在动态规划的部分呈现。

化简容斥原理推导出的式子

鉴于容斥原理本身的性质，如果不对得到的和式进行任何化简，那么求解的复杂度一般至少是 $O(2^{|U|})$ 的。显然，出题人一般不喜欢将这样没有技术含量的做法作为标算，而是更喜欢利用题目的性质化简和式得到复杂度更低的做法。

我们看看之前所做的五道题分别是如何降低复杂度的（可以看出，作为优秀的容斥原理题，它们都没有采用直接计算的方式作为标算）：

- 一道好题：首先观察得到 $f_{\{S_i\}}$ 的取值仅和 $\{|S_i|\}$ 有关，于是改为枚举 $|S_i|$，用组合数计算满足这个条件的 S_i 的数目。进一步的化简将在分治 FFT 的部分呈现。
- 不等关系：对于 f_S，由 S 导出了连续段的长度 $l_1, l_2, \ldots, l_m$，答案仅与 $\frac{1}{\prod_i l_i!}$ 有关。然而，对于不同的 S，导出的 $\{l_i\}$ 仍是不同的。具体的计算方式将在动态规划的部分呈现，并利用分治 FFT 进行加速。
- 氪金手游：对于 f_S，由 S 导出了连通块上的子树 $S_1, S_2, \ldots, S_n$，答案仅与 $\prod_i \frac{p_i}{\sum_{j\in S_i} p_j}$ 有关。然而，对于不同的 S，导出的 $\{S_i\}$ 仍是不同的。具体的计算方式将在动态规划的部分呈现。
- 随机立方体：对于 f_S，有一个性质 q（S 中位置的三维坐标两两不同），当不满足性质 q 时，$f_S = 0$，否则 f_S 仅与 $|S|$ 有关。最终利用枚举 $|S|$ 优化并设法统计满足性质 q 的 S 的个数。

化简容斥原理推导出的式子

鉴于容斥原理本身的性质，如果不对得到的和式进行任何化简，那么求解的复杂度一般至少是 $O(2^{|U|})$ 的。显然，出题人一般不喜欢将这样没有技术含量的做法作为标算，而是更喜欢利用题目的性质化简和式得到复杂度更低的做法。

我们看看之前所做的五道题分别是如何降低复杂度的（可以看出，作为优秀的容斥原理题，它们都没有采用直接计算的方式作为标算）：

- 一道好题：首先观察得到 $f_{\{S_i\}}$ 的取值仅和 $\{|S_i|\}$ 有关，于是改为枚举 $|S_i|$，用组合数计算满足这个条件的 S_i 的数目。进一步的化简将在分治 FFT 的部分呈现。
- 不等关系：对于 f_S，由 S 导出了连续段的长度 $l_1, l_2, \ldots, l_m$，答案仅与 $\frac{1}{\prod_i l_i!}$ 有关。然而，对于不同的 S，导出的 $\{l_i\}$ 仍是不同的。具体的计算方式将在动态规划的部分呈现，并利用分治 FFT 进行加速。
- 氪金手游：对于 f_S，由 S 导出了连通块上的子树 $S_1, S_2, \ldots, S_n$，答案仅与 $\prod_i \frac{p_i}{\sum_{j\in S_i} p_j}$ 有关。然而，对于不同的 S，导出的 $\{S_i\}$ 仍是不同的。具体的计算方式将在动态规划的部分呈现。
- 随机立方体：对于 f_S，有一个性质 q（S 中位置的三维坐标两两不同），当不满足性质 q 时，$f_S = 0$，否则 f_S 仅与 $|S|$ 有关。最终利用枚举 $|S|$ 优化并设法统计满足性质 q 的 S 的个数。
- 曲线救国：对于 $f_{\{S_1,S_2,\ldots,S_k\}}$，有一个性质 q（$\{S_1, S_2, \ldots, S_k\}$ 中集合两两不交），当不满足性质 q 时，$f_{\{S_1,S_2,\ldots,S_k\}} = 0$，否则 $f_{\{S_1,S_2,\ldots,S_k\}}$ 仅与 $\{|S_1|, |S_2|, \ldots, |S_k|\}$ 有关。最终利用枚举 $|S_i|$ 优化并设法统计满足性质 q 的 $\{S_1, S_2, \ldots, S_k\}$ 的个数。最后化简为卷积式并利用 FFT 解决。

化简容斥原理推导出的式子

可以看到，化简式子主要分为两步：

- 分析 f_S 的值是否只和 S 的某些更简单的特征有关，例如 $|S|$ 等。如此一来，我们就可以通过枚举更简单的特征的值，先用组合数计算满足这个特征的 S 的个数，再乘以 f_S。同时要注意，“f_S 的值是否只和 S 的某些更简单的特征有关”也有可能建立在 S 满足一个性质 q 的基础上，在用组合数计算个数时要考虑到这一点。
- 在用第一步化简完式子之后，余下的式子可以结合其它计数算法进行加速，例如利用动态规划、生成函数等。这也是一道综合题的常见解题形式。

容斥通常是题目中的一个难点。不过，只要对容斥背后的原理足够熟悉，清晰地定义每个变量和表达式，及时归纳总结一些容斥方法的应用套路，大家也应该能够对容斥题目更快地上手，培养出扎实的解题能力。

希望本讲的内容对大家有所启发！