min-max 容斥题目选讲

Sergio Gao

2025 年 6 月 14 日

Bingo 模型描述

给定一个 $n \times m$ 的棋盘。
说到棋盘，自然可以定义其行/列。由于它们有类似的结构，我们定义 strip 来统称这两类元素。
strips 之间允许有交。
通过等概率过程让一些数填充到这个棋盘上。
假设一些 strips 构成集合 $STRIPS$。包含所有 strips 的记作 $STRIPS_0$。
对于一种填充方案 $\{b_{nm}\}$，定义最小 bingo 数为：

$$\min_{I \in STRIPS_0} \{\max_{i \in I}\{b_i\}\}$$

求所有填充方案最小 bingo 数的总和。（对于无限过程，则改为求期望）。
把内部套的那层看作 strip 到实数的映射 $G(I) := \max_{i \in I}\{b_i\}$，则由 min-max 容斥，明显下面的式子成立

$$\min_{I\in STRIPS_0}\{G(I)\} = \sum_{\emptyset\neq STRIPS\subseteq STRIPS_0} (-1)^{|STRIPS|-1} \max_{I\in STRIPS}\{G(I)\}$$

考虑这个求和式的局部

$$\max_{I\in STRIPS}\{G(I)\} = \max_{I\in STRIPS}\{\max_{i\in I}\{b_i\}\}$$

很容易发现，他就是 STRIPS 内 strips 交出来的这些元素，求当前填充方案下这些元素的最大值。
由于填充过程是等概率的，这个值往往只和去重后、具体含了多少个元素相关。对于 r 行 c 列，里面显然有 $r\cdot m + c\cdot n - r\cdot c$ 个元素。（这也算一个简单的容斥）
所以我们预处理出 $F(\text{cnt}), \text{cnt} = 1, 2, \cdots, n\cdot m$，然后暴力累加

$$\sum_{r+c\geq 1, r=0,1,\cdots,n, c=0,1,\cdots,m} (-1)^{r+c-1}\binom{n}{r}\binom{m}{c}F(r\cdot m + c\cdot n - r\cdot c)$$

【2024ICPC-Nanjing】I. Bingo

题目描述（简化）

给定两个整数 n、m，以及一个长度为 $n \cdot m$ 的整数序列 $a_1, a_2, \cdots, a_{nm}$，我们要用这些整数填充一个 n 行 m 列的网格。对于一种填充方案 $\{b_{nm}\}$，定义最小 bingo 数为：

$$\min_{\text{strip } I} \left\{ \max_{i \in I} \{b_i\} \right\}$$

请计算所有该序列排列（共 $(nm)!$ 种）下最小 bingo 数的总和。由于答案可能很大，请输出结果对 998 244 353 取模。

这题要预处理的是，选中长度为 $n \cdot m$ 的整数序列中的 cnt 个元素后，遍历所有排列，这些元素最大值的累加。
容易想到这样的做法：不妨把输入的 $\{a\}$ 排序，从头到尾扫一遍，目前扫到 k，数一下有多少种排列这个元素是最右边的。$\binom{k-1}{\text{cnt}-1}$。可以看出 k 从 cnt 遍历到 $n \cdot m$。
请注意 $n \cdot m$ 是常数，后面我们会发现它的锚点作用。
总之

$$F(\text{cnt}) = \sum_{k=\text{cnt}}^{n \cdot m} \binom{k-1}{\text{cnt}-1} a_k$$

简单拆一下组合数，把与 k 无关的因子提出来

$$\frac{F(\text{cnt})}{\text{cnt}!} = \sum_{k=\text{cnt}}^{n \cdot m} (k-\text{cnt})! \times ((k-1)! a_k)$$

这个能高效求吗？

$$\sum_{k=\text{cnt}}^{n\cdot m}(k-\text{cnt})!\times((k-1)!a_k)$$

我特意区分了简单的数值乘法 $\cdot$ 和特殊意味的乘法 $\times$。
看左边，是从 0 开始增到 $n\cdot m-\text{cnt}$。这里 0 是一个锚点。
看右边，是从 cnt 开始增到 $n\cdot m$。这里 $n\cdot m$ 是一个锚点。
what if... 把右边 reverse 成从 0 开始增到 $n\cdot m-\text{cnt}$?
原本同步的 k，右边 reverse 后，就变成一个增 1 一个减 1 了。
枚举下标总和固定，想到了什么？这完全就是两个数组的卷积的某一位！最好能先想象出卷积过程的图形，再符号化成公式

$$(\{N!\}*\{c_N\})_{\text{cnt}}=\sum_{j=0}^{n\cdot m-cnt} j!\times c_{\text{cnt}-j}$$

只需要定义 $c_{n\cdot m-j}=(j-1)!\cdot a_j$ 就等价了。除了提取操作，不包含 cnt，所以 uniformly 用 NTT 预处理一下。

【杭电春季联赛25】宾果游戏

题目描述

一个 $n \times m$ 的表格，玩 bingo 游戏。
每轮随机报出一个位置，在表格中给这个位置做上标记，当某一行/一列上的格子全部被标记时，游戏结束。
表格中有 k 个位置被墨水污染无法做上标记。
请注意，主持人依然会报出被墨水污染的位置，也就是说，每个位置被报到的概率都还是 $\frac{1}{n \times m}$。
请输出游戏结束的期望轮数对 998244353 取模的结果，若游戏无法结束，则输出-1。

这题属于是训练性质的缝合题，我们只列出关键 trick。剩余内容可以从其它题中找到。

题意转化

给每个位置赋予一个最早被标记的时间。
某个 strip I 全被标记的最早时间 $G(I)$，也就是其包含的元素的最大值。某个 strip I 全被标记就结束了，所以就是 $\min\{G(I)\}$。
所以还是在求最小 bingo 数。

此外还需要一个模型

部分抽奖问题 Partial Coupon Collector Problem

对于 N 个等概率发生的事件，单位时间只能发生一个，如果选中 k 个，那么这 k 个全部发生过（也就是最晚的那个发生）的时间期望是

$$\frac{1}{k/N}+\frac{1}{(k-1)/N}+\cdots+\frac{1}{1/N}$$

这可以简单地通过考虑几何分布得到（即期望等于概率的倒数）

背包模型描述

这类问题实际上就是不那么等概率了。于是要根据概率的值域做 dp。还可能需要 NTT 来做加速。

【洛谷 P4707】重返现世

题目描述

有 n 种原料，只需要集齐任意 k 种。每个单位时间，会随机生成一种原料。
每种原料被生成的概率是不同的，第 i 种原料被生成的概率是 $\frac{p_i}{m}$。
如果没有这种原料，那么就可以进行收集。求收集到任意 k 种原料的期望时间，答案对 998244353 取模。
$m \leq 10000$。实际数据保证 $1 \leq p_i$ 且总和为 m。

max 转 min 后，集齐 k 种（全集 S 获得时间第 k 晚）变成得到一种（穷举 $T \subseteq S$ 并分配系数，T 内获得时间第 1 早）。需要用到 k-th min-max，会多一个二项式系数。这题第二个部分也就是处理一个二项式求和。数学事实是：

几何分布

对于一个发生概率为 p 的事件，发生的期望步数为 $1/p$。（根据定义求和然后就是高数习题了）。

处理方法是根据值域 dp（俗称背包问题）（这里所谓值域，观察一下求和式可以想到）。转移方程是二项式系数最简单的递推方程。

吐槽

洛谷题解那里说什么初值很神仙、负负得正，其实完全没那必要。前者其实是因为 min-max 容斥特有的不考虑空集，然而我们 initially 手动处理大小为 1 的集合就行了。后者是有点笨了，硬是要把 $(-1)^k$ 项和二项式下面的一部分绑定起来，然而这玩意和 sigma 枚举的下标无关，直接提到求和号外面就好了。经过测试本题实质上没有 $p = 0$ 的数据。如果有的话已有数据约束其实并不够。

【ABC331】Collect Them All

题目描述

盒子中共有 N 张卡片。每张卡片上写有一个介于 1 和 M（含端点）之间的整数。对于每个 $i = 1, 2, \ldots, M$，恰有 C_i 张卡片上写有数字 i。

从一个空的笔记本开始，你重复以下操作：

- 随机从盒子中抽取一张卡片，将卡片上的数字记入笔记本，然后将卡片放回盒子中。

求直到笔记本中至少写齐所有整数 1 到 M 为止，所执行此操作的次数的期望值，并对结果取模 998244353。

$M \leq N \leq 2 \times 10^5$。

从式子推导来说简直是上个问题的弱化版。但是上个问题跑了背包就得 $O(nm)$ 了，这题过不去。
这里其实也就是优化一下这个背包，把系数放进 generating function，跑多项式乘法，然后再统计答案。
当然这里还有一个类似启发式合并的 trick。多项式乘法允许我们在 $(siz_1 + siz_2)\log(siz_1 + siz_2)$ 合并两个背包，
现在要合理安排合并顺序：即采用一个 priority_queue 维护目前最小的两个背包将它们合并。
区别于经典的启发式合并的每一步扩大一倍，即，只有小量产生贡献而每次贡献完规模至少扩大一倍。这里的分析方式是考虑一个对象合并的遗传链，并把两步合成，看作一组。容易发现，一组合并能使规模至少扩大一倍。即这里大量也产生贡献，但是它也能保证每两步扩大一倍。
这里顺便可以复习一下 NTT。为什么要选原根去取代 FFT 的单位根呢？因为要避免 DFT 时重复代入相同的点值，这会导致 DFT 矩阵出现完全相同的两行，从而不满秩，不可逆，无法保证 IDFT 的正确性。