计数与期望选讲

Wearry

Jan 14, 2020

计数与期望类型的问题在 OI 中属于比较热门的考点，然而严格说来这类问题的考察点非常广，包括但不限于组合计数/DP 技巧/生成函数理论/对概率和期望意义的理解等。

这里总结了常用的方法和知识点，以及一些比较经典的题目的讲解。

常用方法总结

- 组合计数技巧
  - 容斥原理
  - 经典的组合恒等式
- 生成函数
  - 卷积
  - 指数型生成函数
- 概率与期望的性质
  - 条件概率
  - 期望线性性

K Perm Counting

给你两个数 n, k, 询问有多少个长度为 n 的排列 $\{p_i\}$ 满足不存在 $|p_i - i| = k$。

$n \le 2000$

K Perm Counting

由于直接计算不太好处理问题的限制，因此考虑使用容斥原理计算不合法的方案，记 f_i 表示至少有 i 个位置不合法的方案数，那么答案则是：

$$\sum_{i=0}^{n}(-1)^i f_i (n-i)!$$

K Perm Counting

由于直接计算不太好处理问题的限制，因此考虑使用容斥原理计算不合法的方案，记 f_i 表示至少有 i 个位置不合法的方案数，那么答案则是：

$$\sum_{i=0}^{n}(-1)^i f_i (n-i)!$$

于是问题转化成如何求 f，注意到对于某个 $x \in [1,n]$，与之相差 k 的数最多只有两个，因此可以想到将原来的排列拆成若干个互相独立的部分进行 dp 求解答案。

Leftmost Ball

有 n 种颜色的球每种 k 个，现在将这 nk 个球排成一个序列，然后将得到的序列中每种颜色的最左边的一个球染成颜色 0，问这样能够生成出多少种本质不同的序列，对 10^9+7 取模。

$1 \le n, k \le 2000$

Leftmost Ball

通过题意可以知道，所有颜色的球都被分成了两个部分：最左边的一个 0 颜色部分和右边 $k-1$ 个原始颜色部分。

因此在计算方案数的时候也可以考虑将两部分分开处理，从右往左填这个序列，每次要么填上 $k-1$ 个没出现过的颜色的球，要么填上一个已有颜色的 0 球。

Leftmost Ball

通过题意可以知道，所有颜色的球都被分成了两个部分：最左边的一个 0 颜色部分和右边 $k-1$ 个原始颜色部分。

因此在计算方案数的时候也可以考虑将两部分分开处理，从右往左填这个序列，每次要么填上 $k-1$ 个没出现过的颜色的球，要么填上一个已有颜色的 0 球。

记 $f_{i,j}$ 表示已经出现过 i 种颜色，其中 j 种已经填完 0 颜色的方案数，转移的方程也不难得到：

$$f_{i+1,j} \leftarrow f_{i,j}\binom{i(k-1)+j+k-2}{k-2}$$
$$f_{i,j+1} \leftarrow f_{i,j}[i>j]$$

Tournament

求所有 n 个点的竞赛图的强连通分量的数量和，竞赛图指的是对于任意的 $a \neq b$，边 $a \to b$、$b \to a$ 恰好有一条存在的有向图，答案对 10^9+7 取模。

$n \leq 10^5$

Tournament

考虑将一个竞赛图缩点，那么缩完点后的会得到一条“链”，因此竞赛图中强连通分量的个数就是这样的链上边的个数加一。

Tournament

考虑将一个竞赛图缩点，那么缩完点后的会得到一条“链”，因此竞赛图中强连通分量的个数就是这样的链上边的个数加一。

考虑如何求这样的边的数量，通过观察发现，这样的边的数量等于竞赛图中割的数量，根据割的大小进行分类，不难得到计算公式如下：

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

Game with Marbles

袋子里有 r 个红球，g 个绿球和 b 个蓝球，每一轮你从中等概率随机地取出一个球：

- 如果是红球，则扔掉
- 如果是绿球或者蓝球，则将球重新放回袋子里

问恰好拿出过 k 次蓝球的时候，期望下进行了多少轮操作？

$1 \leq r, g, b, k \leq 10^9$

Game with Marbles

可以将期望下进行了多少轮操作的问题替换成等价的，期望下取出了多少个球的问题，同时可以对不同颜色的球分开考虑：

- 蓝球，一定是 k 个。
- 绿球，由于绿球和蓝球的数量都恒定，因此不难理解绿球的期望个数为 $\frac{kg}{b}$ 个。
- 红球，不妨考虑单个红球留下没有被取走的概率，这时其他的红球可以被认为是没有被扔掉而是重新放了回去，因此留下来的概率是 $\left(\frac{b}{b+1}\right)^k$

Game with Marbles

可以将期望下进行了多少轮操作的问题替换成等价的，期望下取出了多少个球的问题，同时可以对不同颜色的球分开考虑：

- 蓝球，一定是 k 个。
- 绿球，由于绿球和蓝球的数量都恒定，因此不难理解绿球的期望个数为 $\frac{kg}{b}$ 个。
- 红球，不妨考虑单个红球留下没有被取走的概率，这时其他的红球可以被认为是没有被扔掉而是重新放了回去，因此留下来的概率是 $\left(\frac{b}{b+1}\right)^k$

最后的答案就是：

$$k + \frac{kg}{b} + r\left(1 - \left(\frac{b}{b+1}\right)^k\right)$$

BBQ Hard

有 n 对 (A_i, B_i)，要求：

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\binom{A_i+B_i+A_j+B_j}{A_i+A_j}$$

$n \le 2\times 10^5, A_i, B_i \le 2000$

BBQ Hard

直接按照定义计算的复杂度是难以承受的，于是不妨考虑式子的组合意义：

BBQ Hard

直接按照定义计算的复杂度是难以承受的，于是不妨考虑式子的组合意义：

- $\binom{A_i+B_i+A_j+B_j}{A_i+A_j}$ 事实上就等于从点 $(0,0)$ 走到点 (A_i+A_j, B_i+B_j) 的只能向右和向上的路径方案数。
- 为方便计算，不妨令 $X_i = (-A_i, -B_i), Y_i = (A_i, B_i)$，那么这一项的值就等于从 X_i 走到 Y_j 的路径方案数。

BBQ Hard

直接按照定义计算的复杂度是难以承受的，于是不妨考虑式子的组合意义：

- $\binom{A_i+B_i+A_j+B_j}{A_i+A_j}$ 事实上就等于从点 $(0,0)$ 走到点 (A_i+A_j, B_i+B_j) 的只能向右和向上的路径方案数。
- 为方便计算，不妨令 $X_i=(-A_i,-B_i), Y_i=(A_i,B_i)$，那么这一项的值就等于从 X_i 走到 Y_j 的路径方案数。
- 由于 A_i, B_i 的值域范围相对比较小，因此直接在网格图上对新的问题模型进行 DP 即可。

Mixed Drinks

给定三个排列 $\{a_i\},\{b_i\},\{c_i\}$，称一个三元组 (x,y,z) 是合法的，当且仅当存在指标集 $S\subseteq[1,n]$，使得：

$$(x,y,z)=(\max_{i\in S}a_i,\max_{i\in S}b_i,\max_{i\in S}c_i)$$

求有合法的三元组的数量。

$n\le 10^5$

Mixed Drinks

对于一个合法三元组，它会对应至少一个下标集合 S。可以发现，对于任意一个合法的三元组，只需要保留不超过最大值所在的不超过 3 个下标构成的指标集即可。

Mixed Drinks

对于一个合法三元组，它会对应至少一个下标集合 S。可以发现，对于任意一个合法的三元组，只需要保留不超过最大值所在的不超过 3 个下标构成的指标集即可。

而且这样的指标集与三元组之间构成一个一一对应的关系，问题转化为求合法指标集的数量:

- 大小为 1 的指标集一定是合法的。

Mixed Drinks

对于一个合法三元组，它会对应至少一个下标集合 S。可以发现，对于任意一个合法的三元组，只需要保留不超过最大值所在的不超过 3 个下标构成的指标集即可。

而且这样的指标集与三元组之间构成一个一一对应的关系，问题转化为求合法指标集的数量：

- 大小为 1 的指标集一定是合法的。
- 大小为 2 的指标集最大值一定分成 $1 + 2$ 的模式，利用二维数点可以计算答案。

Mixed Drinks

对于一个合法三元组，它会对应至少一个下标集合 S。可以发现，对于任意一个合法的三元组，只需要保留不超过最大值所在的不超过 3 个下标构成的指标集即可。

而且这样的指标集与三元组之间构成一个一一对应的关系，问题转化为求合法指标集的数量：

- 大小为 1 的指标集一定是合法的。
- 大小为 2 的指标集最大值一定分成 $1+2$ 的模式，利用二维数点可以计算答案。
- 大小为 3 的指标集与大小为 2 的情形类似，一定有一个指标上取到第一维的最大值，剩下的两个维度可以通过预处理的方式计算。

因此本题可以在 $O(n\log^2 n)$ 的时间复杂度内求出答案。

CF 643E

给出一棵树，初始时只有节点 1，接下来有 q 次操作：

- $1\ fa$，新增一个节点，将父亲设为 fa
- $2\ x$，询问：如果将 x 为根的子树内部的每条边以 $\frac{1}{2}$ 的概率删除，这个子树的期望深度是多少（删除不会影响到之后的操作）

$q \le 5 \times 10^5$，精度误差不超过 10^{-6}。

CF 643E

记 $f_{i,j}$ 表示以 i 为根的子树深度不超过 j 的概率，那么有：

$$f_{i,j} = \prod_{c \in son(i)} \frac{1}{2}(f_{c,j-1} + 1)$$

CF 643E

记 $f_{i,j}$ 表示以 i 为根的子树深度不超过 j 的概率，那么有：

$$f_{i,j} = \prod_{c \in son(i)} \frac{1}{2}(f_{c,j-1} + 1)$$

观察转移，注意到每个层的概率对父亲一层的影响会带上一个 $\frac{1}{2}$ 的系数，所以当转移的层数过多时，影响对于 10^{-6} 级别的误差可以忽略不计，因此只需要记录深度不超过 60 的部分即可。

对于每一操作 1 只需要向上更新 60 层信息即可，复杂度为 $O(60q)$。

USACO 2018-12 T1

有一个标有 $[0, n+1]$ 的数轴，初始时你在位置 x，每一步有两种决策:

- 终止过程，获得 w_x 的收益
- 等概率移动到 $x-1$ 或者 $x+1$，当移动到 0 或 $n+1$ 时直接结束，不获得收益。

对于 $x \in [1, n]$，求从 x 出发时的最大期望收益。

$n \le 10^5, 0 \le w_x \le 10^9$，精度误差 10^{-6}

USACO 2018-12 T1

首先假设在 x 位置选择随机移动，设最大期望收益为 f_x，那么有：

$$f_x = \frac{f_{x-1} + f_{x+1}}{2}$$

否则假设 l, r 是两个相邻的选择终止过程的位置，可以发现：

$$f_l, f_{l+1}, f_{l+2}, \cdots, f_r$$

构成等差数列，于是根据几何意义不难得出最优的解一定在点 (i, w_i) 构成的凸包上。

CERC 2017 G

给出 n 个点 m 条边的无向图，现在从 1 号点出发走向 n 号点。

假设现在在 i 号点买票，买到的票等概率随机通向与 i 相邻的一个点，买到的票可以立即使用或者直接扔掉，扔掉后需要重新买票，求最优决策下期望最少买多少张票能够到达 n 号点。

$n, m \leq 10^5$

CERC 2017 G

记 f_x 表示从 x 出发期望买票的数量，有:

$$f_x = 1 + \frac{1}{deg_x} \sum_{y \in adj(x)} \min\{f_y, f_x\}$$

如果对于 x 有 cnt_x 个相邻节点满足 $f_y < f_x$，那么上式变为:

$$f_x = \frac{deg_x + \sum_{y \in adj(x), f_y < f_x} f_y}{cnt_x}$$

CERC 2017 G

$$f_x = \frac{deg_x + \sum_{y \in adj(x), f_y < f_x} f_y}{cnt_x}$$

考虑首先确定 f_x 最小的 x，然后确定第二小的 x，以此类推。

- 初始时 $f_n = 0$，对于其它点 $f_x = \infty, cnt_x = 0$
- 接下来每次找出当前已经求出的最小的 f_x，对于其他所有相邻的还未确定的节点 y，必有 $f_x < f_y$，因此直接更新 cnt_y, f_y 即可。
- 实现过程类似 Dijkstra 算法的堆优化实现。

LOJ 6509

给定一棵 n 个点的树，点权为 $0/1$，首先随机选择一个点作为起点，然后每次随机移动到一个任意的点并将该点点权异或上 1，求使得所有点点权相同的期望移动距离。

$n \le 10^5$

LOJ 6509

走到一个点 i 时，如果未达到终止条件，就会随机一个点走下去。如果我们知道一个点非最后一次走到的期望次数，乘上这个点到树上其他点的平均距离，再对所有点求和即可得到答案。

不难发现，对于点权相同的点，期望下走到这些点的次数是相同的。不妨记 $dp(i,0/1)$ 表示现在有 i 个 1，点权为 0/1 的点被走到的期望次数，有：

$$dp(i,0) = \frac{i}{n}dp(i-1,0) + \frac{n-i-1}{n}dp(i+1,0) + \frac{1}{n}(dp(i+1,1)+1)$$
$$dp(i,1) = \frac{i-1}{n}dp(i-1,1) + \frac{n-i}{n}dp(i+1,1) + \frac{1}{n}(dp(i-1,0)+1)$$

LOJ 6509

$$dp(i,0) = \frac{i}{n}dp(i-1,0) + \frac{n-i-1}{n}dp(i+1,0) + \frac{1}{n}(dp(i+1,1)+1)$$
$$dp(i,1) = \frac{i-1}{n}dp(i-1,1) + \frac{n-i}{n}dp(i+1,1) + \frac{1}{n}(dp(i-1,0)+1)$$

稍微转化一下式子可以将 $dp(i+1,0/1)$ 表示成 $dp(i,0/1), dp(i-1,0/1)$ 的组合，又因为 $dp(0,0/1)=0$ 因此可以将任意的 $dp(x,0/1)$ 表示成 $dp(1,0/1)$ 的组合，最后利用 $dp(n,0/1)=0$ 可以解出所有的点的答案。

CF446 D

一个 n 个点 m 条边的无向图，有些点上有一个陷阱，n 号点上一定有一个陷阱。从 1 号点出发，每次等概率随机走向一个相邻的点。初始时有 k 的生命值，如果走到一个有陷阱的点，就会丢失 1 的生命值。如果在恰好有 2 生命值时走进 n 号点，就算游戏成功。求游戏成功的概率。

$n \leq 500, m \leq 10^5, k \leq 10^9, a \leq 101$，其中 a 是有陷阱的点的数量，精度要求为 10^{-4}。

CF446 D

定义关键点为有陷阱的点，对于每一个关键点 v，定义 $w(u,v)$ 为：从 u 出发遇到的第一个关键点恰好是 v 的概率。

在得到这样的矩阵之后可以利用矩阵乘法优化 DP 的方式在 $O(a^3 \log k)$ 的时间内求出原问题的答案，问题的难点在于如何求 $w(u,v)$。

CF446 D

定义关键点为有陷阱的点，对于每一个关键点 v，定义 $w(u,v)$ 为：从 u 出发遇到的第一个关键点恰好是 v 的概率。

在得到这样的矩阵之后可以利用矩阵乘法优化 DP 的方式在 $O(a^3 \log k)$ 的时间内求出原问题的答案，问题的难点在于如何求 $w(u,v)$。

暴力的做法是考虑对于每个 v，都直接使用高斯消元法计算 $w(u,v)$，然而这样的时间复杂度是难以承受的。

注意到高斯消元法的本质是求解 $A\mathrm{x}=\mathrm{v}$ 这样形式的方程组，而在本题中 A 矩阵是一直不变的，变化的是向量 v，于是可以首先用 $O(n^3)$ 的时间求出 A^{-1} 然后对于不同的 v，只需要求 $A^{-1}\mathrm{v}$ 即可得到方程的解。

CF183 D

有 n 个人和 m 中尺寸的衣服，每个人只适合一个尺寸的衣服，现在你只知道第 i 个人适合尺寸 j 的概率。

现在你可以准备 n 件任意尺寸的衣服，要求最大化收到合适衣服的人的数量的期望。

$n \leq 3000, m \leq 300$

CF183 D

令 $f_{i,j}$ 表示有至少 j 个人适合尺寸 i 的概率，将最大的 n 个 $f_{i,j}$ 求和就得到了问题的答案。

直接计算所有的 $f_{i,j}$ 复杂度较高，注意到对于固定的 i，$f_{i,j}$ 严格单调，于是每次维护某个 i 的一个头指针，取出最大的一个然后求取出的那一项的下一项即可。

复杂度 $O(n^2)$

AGC032 F

有一个面积为 1 的圆盘，现在会进行 n 次切割。每次切割会随机圆上的一个点，将圆心到圆上这一点之间切开，得到若干个扇形区域。

在经过随机的 n 次切割后，求某一个连续的扇形区域与 $\frac{1}{3}$ 圆之间的最小面积差的期望大小。

$n \leq 10^6$，答案对 $10^9 + 7$ 取模。

AGC032 F

不妨假设第一次切割的位置为 0，接下来如果在 α 的位置切开了一次，则在 α 的位置标一条红线，$\alpha+\frac{2\pi}{3}$ 的位置表一条绿线，$\alpha+\frac{4\pi}{3}$ 的位置标上一条蓝线。

这样问题就转化成了，在 $[0,\frac{2\pi}{3}]$ 的区间内，相距最近的两根不同颜色的线的夹角是多少。

AGC032 F

不妨假设第一次切割的位置为 0，接下来如果在 α 的位置切开了一次，则在 α 的位置标一条红线，$\alpha+\frac{2\pi}{3}$ 的位置表一条绿线，$\alpha+\frac{4\pi}{3}$ 的位置标上一条蓝线。

这样问题就转化成了，在 $[0,\frac{2\pi}{3}]$ 的区间内，相距最近的两根不同颜色的线的夹角是多少。

区间在经过所有操作之后一定会被分成 n 块，又因为划分是完全随机的，根据相关结论有：第 i 小的面积是 $\sum_{j=1}^{i}\frac{1}{3n(n-j+1)}$。

同时颜色也是完全随机的，因此每一段的颜色有 $\frac{1}{3}$ 的概率相同，简单整理后就得到：

$$E(angle)=\sum_{i=1}^{n}\frac{1}{3^{i-1}}\times\frac{1}{3n(n-i+1)}$$