线性代数专题分享

Little09

2025年8月3日

前言

这次分享的内容为一些 OI 里的数学相关内容，总体难度不大。
题目中会包含一些简单题、经典题，想必在座的同学有很多见过，不过拿出来讲一是考虑一些没见过对应套路的同学，二是见过的同学也可以加深一下影响。
希望大家都能有所收获!

矩阵乘法

矩阵相乘只有在第一个矩阵的列数和第二个矩阵的行数相同时才有意义。
设 A 为 $P \times M$ 的矩阵，B 为 $M \times Q$ 的矩阵，设矩阵 C 为矩阵 A 与 B 的乘积，
其中矩阵 C 中的第 i 行第 j 列元素可以表示为：

$$C_{i,j} = \sum_{k=1}^{M} A_{i,k} B_{k,j}$$

矩阵乘法满足结合律，不满足一般的交换律。
按照定义计算一次矩阵乘法复杂度为 $O(PMQ)$，同阶时即为常说的 $O(n^3)$。

矩阵快速幂

利用结合律，可以在 $O(n^3 \log m)$ 的时间复杂度内计算矩阵的 m 次幂。
由于线性递推式可以表示成矩阵乘法的形式，也通常用矩阵快速幂来求线性递推数列的某一项。

矩阵的秩

我们对一个向量组 $a_1, \ldots, a_n$ 定义线性相关，如果方程组 $\sum_{i=1}^{n} k_i a_i$ 存在非全零的解，则称这组向量线性相关。
线性相关可以理解为「多余」，说明向量组内部有的向量可以被其他向量表出。
把多余的向量删去，删完了之后，将剩下极大线性无关组。一组列向量的秩为极大线性无关组的大小。从向量组删向量的删法不唯一，因此极大线性无关组也不唯一。
对矩阵而言，秩即为列向量的秩。可以证明也等于行向量的秩。

矩阵加速递推

$$f_1 = f_2 = 0$$
$$f_n = 7f_{n-1} + 6f_{n-2} + 5n + 4 \times 3^n$$

如果矩阵仅有这三个元素 $\begin{bmatrix} f_n & f_{n-1} & n \end{bmatrix}$ 是难以构造出转移方程的，因为乘方运算和 $+1$ 无法用矩阵描述。
于是考虑构造一个更大的矩阵。

$$\begin{bmatrix} f_n & f_{n-1} & n & 3^n & 1 \end{bmatrix}$$

我们希望构造一个递推矩阵可以转移到

$$\begin{bmatrix} f_{n+1} & f_n & n+1 & 3^{n+1} & 1 \end{bmatrix}$$

转移矩阵就容易写出。

动态 DP

给定一个 n 个点带点权的树，支持 m 次修改点权，并查询最大独立集。
$n, m \leq 10^5$。

动态 DP

首先最大独立集问题是一个简单的树形 DP，求一次可以 $O(n)$ 做，记 $dp_{i,0/1}$ 表示第 i 个节点的子树内，是否选择第 i 个节点的最大点权和。那我们有很简单的方程：

$$f_{i,0} = \sum_j \max(f_{j,0}, f_{j,1})$$

$$f_{i,1} = (\sum_j f_{j,0}) + a_i$$

我们希望在通过树链剖分优化修改后的 DP 转移，于是我们可以定义 g 数组：$g_{i,1}$ 表示所有轻儿子都不选的最大权独立集再加 a_i，$g_{i,0}$ 表示所有轻儿子可选可不选的最大权独立集。于是我们得到新的转移式：

$$f_{i,0} = g_{i,0} + \max(f_{j,0}, f_{j,1})$$

动态 DP

这个式子其实反应出了我们应当怎么理解 g 这个数组，就是在正常的 DP 转移中我们需要通过 i 节点的每个儿子进行转移，而现在我们把所有非重儿子相关的项取出来构成 g（这里包括轻儿子和自己），其余项保留，这样的构成有利于我们接下来配合树链剖分。

注意到上面这个式子很矩阵乘法，那我们把它写成矩阵的形式。

我们定义广义矩阵乘法是 $C_{i,j} = \max_{k=1}^{n} A_{i,k} + B_{k,j}$，那么有：

$$\begin{bmatrix} f_{s_i,0} & f_{s_i,1} \end{bmatrix} \times \begin{bmatrix} g_{i,0} & g_{i,1} \\ g_{i,0} & -\infty \end{bmatrix} \to \begin{bmatrix} f_{i,0} & f_{i,1} \end{bmatrix}$$

当然应该转化一下，写成把 g 放在左侧的形式。

动态 DP

我们初始的时候先预处理 g 数组，在动态 DP 的过程中我们动态地维护 g 数组。于是我们如果需要查询 f 数组，我们只需要查询它到链底的一串 g 的矩阵乘积就行。不过这一点基于对于叶子节点 $f_i = g_i$。
接下来我们考虑修改操作。我们发现一次修改仅修改 $O(\log n)$ 个 g 值，因为只有在链顶端的时候对应的是父亲节点的轻儿子，这个时候需要修改 g，其余时候显然 g 是不变的。
接下来就是一个关键问题：如何在修改完 i 的轻儿子的 f 值后修改 i 这个点的 g 值。注意到 g 中由轻儿子的值的和构成，这个满足可减性，于是计算出原先这个轻儿子的 f 值，减掉原来的值再加上新的值就行。
需要注意更新 g 时先不在线段树上修改，因为我们还需要求出原先的 f 值。

定长路径统计 / 定长最短路

用邻接矩阵表示图，做矩阵快速幂。

线性基

如果有 n 个空间向量，不存在不全为 0 的序列 a 满足：

$$\sum_{i=1}^{n} a_i v_i = 0$$

则称这 n 个向量线性无关，否则为线性相关。一组线性无关的向量组可以称为线性基。
下文讨论异或线性基。对异或线性基，条件即为不存在 XOR 为 0 的非空子集。

构造线性基

对原集合的每个数 p 转为二进制，从高位向低位扫，对于第 x 位是 1 的，如果 a_x 不存在，那么令 $a_x \leftarrow p$ 并结束扫描，如果存在，令 $p \leftarrow p \operatorname{xor} a_x$。
查询原集合内任意几个元素 xor 的最大值，只需将线性基从高位向低位扫，若 xor 上当前扫到的 a_x 答案变大，就把答案异或上 a_x。
为什么能行呢？因为从高往低位扫，若当前扫到第 i 位，意味着可以保证答案的第 i 位为 1，且后面没有机会改变第 i 位。
查询原集合内任意几个元素 xor 的最小值，就是线性基集合所有元素中最小的那个。
查询某个数是否能被异或出来，类似于插入，如果最后插入的数 p 被异或成了 0，则能被异或出来。

线性基求并

把一个线性基的向量取出来插入另一个线性基即可。
复杂度为 $O(\log^2 v)$。

P4151 [WC2011]最大XOR和路径

给定一个带权无向图，求 1 到 n 的 xor 最长路。

$n, m \leq 10^5$。

P4151 [WC2011]最大XOR和路径

给定一个带权无向图，求 1 到 n 的 xor 最长路。
$n, m \leq 10^5$。

考虑一条路径是怎么构成的。我们先搞出来原图生成树，考虑返祖边构成的环。可以证明所有路径的 xor 可以由 $1 \to n$ 的 xor 拼上任意个返祖边构成的环构成。所以对环建线性基。

P3292 [SCOI2016]幸运数字

给定 n 个点带点权的树，q 次询问一条路径的集合的子集中 xor 最大值。

$n \leq 2 \times 10^4$，$q \leq 2 \times 10^5$。

P3292 [SCOI2016]幸运数字

给定 n 个点带点权的树，q 次询问一条路径的集合的子集中 xor 最大值。
$n \le 2 \times 10^4$，$q \le 2 \times 10^5$。

首先有一个很 naive 的想法是树剖维护线性基，这样喜提四个 log。我们考虑线性基可以有重复计算的地方（类似 ST 表），那么用倍增维护，可以 $O(n \log^2 v \log n + q \log^2 v)$。
注意到这个问题好像和树上路径背包问题很像，都是插入 log，合并 $\log^2$，那考虑点分，每个分治重心处理出每个点到重心的线性基，这一步是 $O(n \log n \log v)$ 的。接下来每个经过重心的询问，我们就可以用两个线性基并一下了，复杂度
$O(n \log n \log v + q \log^2 v)$。

CF1100F Ivan and Burgers

给定一个序列，q 次询问，每次给定区间 $[l,r]$，求从中选出一些数 xor 最大。
$n,q \le 5\times 10^5$。

CF1100F Ivan and Burgers

给定一个序列，q 次询问，每次给定区间 $[l, r]$，求从中选出一些数 xor 最大。
$n, q \le 5 \times 10^5$。

扫描右端点，在线性基里的每个位置记一下插入的时间。如果当前插进来的比原先的新，就把当前的给放进去，把原先的拿出来 xor 一下再滚下去。
这样 $[l, r]$ 的线性基就是在 r 的时候的线性基，取出所有 $\ge l$ 的元素构成的线性基。

P3292 [SCOI2016]幸运数字

给定 n 个点带点权的树，q 次询问一条路径的集合的子集中 xor 最大值。

$n \le 2 \times 10^4$，$q \le 2 \times 10^5$。

P3292 [SCOI2016]幸运数字

给定 n 个点带点权的树，q 次询问一条路径的集合的子集中 xor 最大值。
$n \leq 2 \times 10^4$，$q \leq 2 \times 10^5$。

考虑用前缀线性基做，每个点在父亲的基础上插入，优先级是深度。这样查询的时候就是搞出两个线性基合并，复杂度可以变成 $O(n \log v + q \log^2 v)$。

LOJ 6040. 「雅礼集训 2017 Day5」矩阵

给定一个 $N \times N$ 的 01 矩阵 C，求有多少个 $N \times N$ 的 01 矩阵 A, B 满足 $A \times B = C$。运算均在 mod2 意义下进行。
$N \leq 2000$。

LOJ 6040. 「雅礼集训 2017 Day5」矩阵

我们把 A, C 的每一行看做一个 n 维向量，那么 $B_{i,j}$ 就表示 A_i 是否要插入到 C_j 去。假设已经确定了矩阵 A，那么对于 C_j，首先其一定需要可以被 A 线性表示出来。如果 C 的每一行都满足条件，那么根据线性基相关知识，B 的总方案数是 $(2^{n-r})^n$，其中 r 表示 A 的秩。

LOJ 6040. 「雅礼集训 2017 Day5」矩阵

我们发现一个性质，由于 A 覆盖了所有的情况，因此对于秩相同的 C，其对应的答案必然是一样的。那么我们可以计算秩为 r 的 C 的答案和。

计算 $dp_{i,j}$ 表示 $n \times i$ 的矩阵，其秩为 j 的方案数。这个 DP 容易递推，我们每次新加一列，秩最多增加 1，只需要计算秩不变的方案数即可。那么我们枚举 A 的秩 x，此时我们对满足条件的 C 的每一行可以视为一个长度为 x 的 01 向量，且这个 $n \times x$ 的矩阵的秩等于原先矩阵 C 的秩。所以这样的 A, C 对的方案数为 $f_{n,x} \times f_{x,r}$。

高斯消元

高斯消元就是将一个普通矩阵不断消成上三角矩阵，甚至对角矩阵。
三种初等行变换：交换两行，某一行乘非零 k，某一行加上另一行乘 k。
容易发现进行这三种初等行变换解不变。
依次考虑每个未知数，找到和这个未知数相关的方程。然后把其他方程里的该未知数消掉即可。这样可以得到一个上三角矩阵。
如果消的时候直接在所有方程里消元，就可以得到对角线矩阵（若满秩）。

矩阵求逆

给定一个 $n \times n$ 的矩阵求逆。并判断是否存在逆。逆矩阵定义为 $A^{-1}A = AA^{-1} = I$，其中 I 为单位矩阵。运算对 $10^9 + 7$ 取模。

矩阵求逆

给定一个 $n \times n$ 的矩阵求逆。并判断是否存在逆。逆矩阵定义为 $A^{-1}A = AA^{-1} = I$，其中 I 为单位矩阵。运算对 $10^9 + 7$ 取模。

不妨 $A^{-1} = B$。考虑高斯消元是一个线性变换的过程，将 A 进行线性变换相当于左乘一个矩阵。考虑 $BA = I$，$BI = B$，也就是说我们把 A, I 放在一起做线性变换（高斯消元），当 A 变成 I 的时候，I 也就变成了 B。
如果秩不为 n（消元的时候某行找不到主元），那么也就无法变成 I，也就是不存在逆。

P2447 [SDOI2010] 外星千足虫

有 m 个异或方程组，找到最小的 k 满足前 k 个方程组即可解出所有未知数，或声称总有未知数无法确定。

P2447 [SDOI2010] 外星千足虫

有 m 个异或方程组，找到最小的 k 满足前 k 个方程组即可解出所有未知数，或声称总有未知数无法确定。

首先异或方程组是可以用 bitset 优化的，一般复杂度 $O(\frac{n^3}{w})$。但这道题还有找到 k，如果二分的话就太拉胯了。考虑在消元的时候，对每个元要往后找到第一个方程组，直接对这个取 max 即可。

随机游走

给定一张 n 个点的有向图，从 1 号点出发，每次等概率选一条出边走，问期望多少步能走到 n 号点。
$nd300$。

随机游走

给定一张 n 个点的有向图，从 1 号点出发，每次等概率选一条出边走，问期望多少步能走到 n 号点。
$nd300$。

令 f_i 表示从 i 出发期望多少步能到 n 号点，转移是：

$$f_u = \frac{\sum_{u \to v} f_v}{deg_u} + 1$$

使用高斯消元解这个方程组即可。

[HNOI2013]游走

给定一个 n 个点 m 条边的无向连通图，顶点从 1 编号到 n，边从 1 编号到 m。小 Z 在该图上进行随机游走，初始时小 Z 在 1 号顶点，每一步小 Z 以相等的概率随机选择当前顶点的某条边，沿着这条边走到下一个顶点，获得等于这条边的编号的分数。当小 Z 到达 n 号顶点时游走结束，总分为所有获得的分数之和。现在，请你对这 m 条边进行编号，使得小 Z 获得的总分的期望值最小。

$2 \le n \le 500$，$1 \le m \le 125000$，$1 \le u, v \le n$，给出的图无重边和自环，且从 1 出发可以到达所有的节点。

[HNOI2013]游走

我们可以算出每条边期望被经过的概率，然后从大到小分配从小到大的编号即可。那么问题就是怎么算出概率。我们考虑先统计点的期望次数。

这又是一个经典问题。定义 f_i 表示第 i 个点的期望经过次数，对于一般的点，有：

$$f_u = \sum_{(u,v)\in G} \frac{f_v}{deg_v}$$

注意如果 $v = n$，那是不计入的，因为到达 n 以后不会再走。以及 f_1 需要在此基础上 $+1$，因为初始位置在 1。

接下来高斯消元可得 f。再考虑边也是简单的，对于 (u, v)，考虑 $u \to v$ 和 $v \to u$ 的期望次数即可。那么就是：

$$E_{(u,v)} = \frac{f_u}{deg_u} + \frac{f_v}{deg_v}$$

行列式

行列式有公式

$$\det(A) = \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} a_{i,\sigma(i)}$$

其中 S_n 是指长度为 n 的全排列的集合，σ 就是一个全排列，如果 σ 的逆序对对数为偶数，则 $\operatorname{sgn}(\sigma) = 1$，否则 $\operatorname{sgn}(\sigma) = -1$。
类似地，积和式的定义式为：

$$\operatorname{perm}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i,\sigma(i)}$$

行列式

一些代数初等变换对 det 的影响：

1. 矩阵转置，行列式不变；
2. 矩阵行（列）交换，行列式取反；
3. 矩阵行（列）相加或相减，行列式不变；
4. 矩阵行（列）所有元素同时乘以数 k，行列式等比例变大。

行列式

一些简单性质：

$\det(AB) = \det(A)\det(B)$

mod2 意义下就是积和式（二分图匹配计数 mod2）

$\det(A+B)$：用定义展开一下，得到 2^n 种展开，每行在 A 或 B 中选，求和。

行列式求值

对矩阵应用高斯消元之后，我们可以得到一个对角线矩阵，此矩阵的行列式由对角线元素之积所决定。其符号可由交换行的数量来确定（如果为奇数，则行列式的符号应颠倒）。因此，我们可以在 $O(n^3)$ 的复杂度下使用高斯算法计算矩阵。
注意，如果在某个时候，我们在当前列中找不到非零单元，则算法应停止并返回 0。
当 mod 不是质数的时候没有逆元，我们需要考虑对两行（列）做辗转相减消元。

矩阵树定理

以下为无向图版本的矩阵树定理。
定义 Laplace 矩阵 $L = D - A$，其中 D 是度数矩阵，A 是邻接矩阵，即对于每条边 (i, j)，将 $L_{i,i}$ 和 $L_{j,j}$ 加一，将 $L_{i,j}$ 和 $L_{j,i}$ 减一。
定义 $t(G)$ 表示这张图的生成树，有：

$$t(G) = \det L(G)\begin{pmatrix} 1, 2, \cdots, i-1, i+1, \cdots, n \\ 1, 2, \cdots, i-1, i+1, \cdots, n \end{pmatrix}$$

表示把 L 删掉其中的任意第 i 行和第 i 列。无向图的 Laplace 矩阵所有 $n-1$ 阶主子式都相等。

矩阵树定理

对于有向图版本的矩阵树定理：
同理定义 Laplace 矩阵 $L = D - A$。但是 $L^{out} = D^{out} - A$，这里 D^{out} 表示出度矩阵。L^{in} 同理。
以 r 为根的内向树方案，只需要在 L^{out} 上去掉 r 行 r 列后求行列式。
以 r 为根的外向树方案，只需要在 L^{in} 上去掉 r 行 r 列后求行列式。

P6624 [省选联考 2020 A 卷] 作业题

给定带权无向图 G，求所有生成树 T 所包含的边的边权的最大公约数乘以边权之和。

$n \le 30$，$w_i \le 1.5 \times 10^5$。

P6624 [省选联考 2020 A 卷] 作业题

给定带权无向图 G，求所有生成树 T 所包含的边的边权的最大公约数乘以边权之和。
$n \le 30$，$w_i \le 1.5 \times 10^5$。

枚举 gcd，容斥，变成枚举其中一个因数，然后只保留包含这个因数的。考虑矩阵树定理。但是做权值积的和是可以做，权值和不好做。考虑把每个权值当成一个一次多项式 $cx + 1$，放在 $\bmod x^2$ 意义下，那么这些多项式的乘积的一次项就是权值和。所以我们把它转化成了权值积，定义一下多项式的四则运算，正常求行列式即可。但是注意到由于加加减减，做除法的时候不一定有逆元。也就是一次项可能为 0，这个时候我们需要在高斯消元的时候找一个不为 0 的项，如果所有项都为 0 那就是常数除法，也没问题。

GYM104725C 国王的疑惑

给出一张 n 个点 m 条边的有向图，将这个图复制成 K 份，求整张图补图的以 1 为根的外向生成树的个数。

$n \leq 300, k \leq 10^8$。

GYM104725C 国王的疑惑

首先可以考虑暴力一点，用矩阵树定理写出整个矩阵。第 i 个点的度数 d_i 是很好算了。然后我们尝试表示出最大的矩阵。我们发现我们可以把最大的矩阵划分成 K^2 个小矩阵，位于对角线上的矩阵是相等的，也就是原图对应的矩阵；其他矩阵所有元素均为 -1。那么我们的目标就是求这个大矩阵去掉第一行第一列后的行列式。

注意到 $d_i = \sum A_{i,j}(i \neq j)$，所以我们如果把大矩阵的每一行加到最后一行，最后一行就会变成全 0。那对于删去第一行第一列的余子式，我们在最后一行会得到：

$$[A_{1,2}, A_{1,3}, ..., A_{1,n}, 1, 1, 1, ..., 1]$$

再把这一行加到其他行，我们发现对角线以上的 -1 都被消掉了，那么这个矩阵就类似是一个上三角矩阵了。我们只要对对角线上的矩阵求行列式就行。我们发现除了第一个和最后一个，其他矩阵是相等的。所以求完后计算快速幂就行。

BEST 定理

设 G 是有向欧拉图，那么 G 的不同欧拉回路总数 $ec(G)$ 是

$$ec(G) = t^{root}(G,k)\prod_{v\in V}(\deg(v)-1)!$$

$t^{root}(G,k)$ 表示 G 中以 k 为根的内向树的个数，可以用矩阵树定理计算。
注意，对欧拉图 G 的任意两个节点 k,k'，都有 $t^{root}(G,k)=t^{root}(G,k')$，且欧拉图 G 的所有节点的入度和出度相等。

牛客挑战赛 43 F 矩阵与排列

有 n 个 3×3 的矩阵 $A_1,\dots,A_n$，求对于所有排列 p，计算 $(A_{p_1}\times A_{p_2}\times .. \times A_{p_n})_{1,1}$ 之和模 998244353 的值。
$n\le 40$。

牛客挑战赛 43 F 矩阵与排列

考虑矩阵乘法的意义，把每个矩阵看做有向图的邻接矩阵，那么这个的实际意义就是要在每张无向图中选择一条边，再选择一个排列，使得按照这个排列，这些边恰好构成一条从 1 号点出发的回路。即在每张无向图中选择一条边后，答案要乘上这张图的欧拉回路个数。由于要算欧拉回路个数，考虑 BEST 定理，如果为欧拉图那么欧拉回路个数是 $T \times \prod_{i=1}^{n}(deg_i - 1)!$，其中 T 表示图的外向（或内向）生成树的个数。在本题中由于第一条边不固定，所以还要乘 deg_1。

牛客挑战赛 43 F 矩阵与排列

那么我们要求的答案就是对于每张图选择一条边，满足所有点的入度和出度相等，再在选出的边中选择一个边集构成一棵外向树，再乘上 $deg_1 \times \prod_{i=1}^{n}(deg_i - 1)!$。注意到目前选择的边集是可以压起来的，只有 6 个状态，于是我们设 $dp(i, x_1, y_1, x_2, y_2, G)$ 表示目前做到第 i 个点，当前 1 号点的入度出度为 x_1, y_1，2 号点的入度出度为 x_2, y_2，目前选入外向树的边集是 G 的方案数。欧拉回路中存在和 1 号点不连通的孤立点是合法的，最后统计答案时应考虑到这种情况。这样预处理转移然后直接 DP 的复杂度是 $O(n^5)$。

LGV 引理

可以解决有向无环图上不相交路径计数，核心在于考察行列式中逆序对数的组合意义。

假设 G 为有限的带权有向无环图，有起点集合 $A=\{a_1,\dots,a_n\}$ 和终点集合 $B=\{b_1,\dots,b_n\}$。对于一条路径 P，定义权值为经过的所有边权之积。定义 $e(a,b)$ 表示 $a\to b$ 所有路径的权值的和。

定义矩阵 M 满足 $M_{i,j}=e(a_i,b_j)$。一组 $A\to B$ 的不相交路径 S：S_i 是一条从 A_i 到 $B_{\sigma(S)_i}$ 的路径（$\sigma(S)$ 是一个排列），对于任何 $i\neq j$，S_i 和 S_j 没有公共顶点。$N(\sigma)$ 表示排列 σ 的逆序对个数。

$$\det(M)=\sum_{S:A\to B}(-1)^{N(\sigma(S))}\prod_{i=1}^{n}\omega(S_i)$$

简要证明：对于出现交点的路径，接下来两边走法对称，导致算逆序对的时候 -1 和 $+1$ 相抵消。

[NOI2021] 路径交点

有一个 k 层的有向图，第 i 层有 n_i 个点，$n_1 = n_k$，相邻两层之间有连边。现在选出 n_1 条路径，从第一层到第 k 层，要求每个点只经过一次。对于整个路径方案，它的交点数为两两不同路径间交点数之和。求有偶数个交点的路径方案数比有奇数个交点的路径方案数多多少个。

$k, n \leq 100$。

[NOI2021] 路径交点

有一个 k 层的有向图，第 i 层有 n_i 个点，$n_1 = n_k$，相邻两层之间有连边。现在选出 n_1 条路径，从第一层到第 k 层，要求每个点只经过一次。对于整个路径方案，它的交点数为两两不同路径间交点数之和。求有偶数个交点的路径方案数比有奇数个交点的路径方案数多多少个。
$k, n \leq 100$。

观察到两条路径的交点的奇偶性只取决于起点和终点是否形成逆序对。
该条件与 LGV 引理所述相同，因此只要求出每个起点到每个终点的路径条数。
即把所有矩阵相乘后计算行列式。

[ABC216H] Random Robots

有 K 个机器人在数轴上，位置分别是 $x_1, x_2, \ldots, x_K$，x 均为整数。接下来 n 秒，每秒每个机器人有 $\frac{1}{2}$ 的概率不动，$\frac{1}{2}$ 的概率往坐标轴正方向移动一个单位距离，机器人的移动同时进行。求机器人互相不碰撞的概率，对 998244353 取模。

$2 \le K \le 10$，$1 \le n \le 1000$，$0 \le x_1 < x_2 < \ldots < x_K \le 1000$。

[ABC216H] Random Robots

把时间轴看做 y 坐标，那么发现每个机器人独立从 $y=0$ 走到 $y=n$，每步向上或右上走一步，求路径没有交点的方案数。然而终点并不知道，不能直接套用 LGV 引理。考虑假设枚举了终点集合 $b_1,\dots,b_k$，答案就是

$$\sum_p (-1)^{\mathrm{sgn}(p)} \prod A_{i,p_i}$$

其中 $A_{i,j}$ 表示从 $(0,x_i)$ 走到 (n,b_j) 的方案数。接下来我们对这个东西做状压 DP，考虑定义 $dp_{x,S}$ 表示目前做到坐标 (n,x)，此时已经被使用的初始位置机器人集合是 S 的贡献，每次要么啥也不干，要么枚举一个初始位置的机器人走到这里加入贡献。

CF1864H Asterism Stream

你初始有一个数 x，每秒钟会随机将其 $+1$ 或 $\times 2$，问期望几秒后 x 不小于 n。T 组询问。
$T \le 100$，$n \le 10^{18}$。

CF1864H Asterism Stream

首先倒过来考虑可以得到一个很简单的 $O(n)$ 的 DP：
$f(n) = 1 + \frac{1}{2}(f(n-1) + f(\lceil \frac{n}{2} \rceil))$，边界条件 $f(1) = 0$。注意到式子里有个向上取整，我们把它换成向下取整：定义
$g(n) = f(n+1)$，可以得到 $g(n) = 1 + \frac{1}{2}(f(n-1) + f(\lfloor \frac{n}{2} \rfloor))$，边界条件 $g(0) = 0$。这样转化后最后答案就是 $g(n-1)$。

CF1864H Asterism Stream

考虑用矩阵加速递推。我们维护向量 $b_n = [1, g(n), g(\lfloor \frac{n}{2} \rfloor), g(\lfloor \frac{n}{4} \rfloor), ...]$。想一下怎么从 b_{n-1} 转移到 b_n：依次考虑每一位，我们发现假设 n 在二进制表示下最后有 k 个 0，那么在 1 后面的 $k+1$ 个位置的转移是用 b_{n-1} 的对应位置和 b_n 的下一个位置的值相加，再除以 2 再加 1；而这 $k+1$ 个数以后的数只需要直接从 b_{n-1} 对应位置转移过来就行，因为这些数是完全相同的。所以这个转移矩阵只与 k 相关。所以我们对每个 k 预处理出这个矩阵，记作 M_k。

CF1864H Asterism Stream

接下来考虑对于我们需要求的 $g(n)$，我们只需要计算 $\prod_{i=1}^{n} M_{\mathrm{ctz}(i)}$，其中 $\mathrm{ctz}(i)$ 表示 i 的二进制表示末尾有几个 0。这个问题我们只需要对每个 2^k 求出 $\prod_{i=1}^{2^k} M_{\mathrm{ctz}(i)}$，每次询问的时候计算 $\log n$ 次向量乘矩阵就行。
复杂度 $O(\log^4 n + T \log^3 n)$。

CF468E Permanent

给定一个 $n \times n$ 的矩阵，初始时矩阵的每个元素都为 1。又给出 k 个三元组 (x, y, z)，表示将矩阵的 (x, y) 位置变成 z。你需要求出完成这些变换之后矩阵的积和式。对 $10^9 + 7$ 取模。

积和式的定义：

$$per(A) = \sum_{\pi} \prod_{i=1}^{n} a_{i,\pi(i)}$$

其中 π 取遍全部 $1 \sim n$ 共 $n!$ 个排列。

$n \le 10^5, k \le 50$。

CF468E Permanent

考虑积和式的本质等价于二分图完美匹配计数，所以我们可以把二分图搞出来。
注意到大部分的边都是 1，所以我们考虑最终匹配里的特殊边，考虑把每条边权值减 1，那么就可以转化为：对每个 t，计数匹配的 t 组的方案数，乘以 $(n-t)!$，因为剩下 $n-t$ 组可以任意匹配。

CF468E Permanent

那么注意到我们现在可以对每个连通块考虑，最后做暴力卷积就行：对于一个点数为 n 边数为 m 的连通块，如果 n 比较小，我们取左右两边较小的一边做状压；如果 $m-n$ 比较小，我们考虑取出生成树，枚举非树边是否选择，再在树上做 DP。
综合以上两个算法，可以得到复杂度 $O(k^2 2^{\frac{k}{3}})$。

结语

谢谢大家！