线性代数选讲

徐骁扬

线性方程组和矩阵

矩阵的引入来源于线性方程组，例如对于线性方程组：

$$\begin{cases} x_1 & +2x_2 & -x_3 & = 2 \\ 2x_1 & +x_2 & +x_3 & = 7 \\ -x_1 & +3x_2 & & = 5 \end{cases}$$

将上面的系数和未知数分开，可以被写成矩阵运算的性质：

$$\begin{bmatrix} 1 & 2 & -1 \\ 2 & 1 & 1 \\ -1 & 3 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 7 \\ 5 \end{bmatrix}$$

矩阵

由 $m \times n$ 个数 a_{ij} 排成的 m 行 n 列的数表：

$$\begin{matrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{matrix}$$

称为 m 行 n 列的矩阵，简称 $m \times n$ 矩阵。

矩阵

为表示它是一个整体，可以在其两侧加一对小括号或中括号，并用大写字母表示，记作：

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

这 $m \times n$ 个数称为矩阵 A 的元素，简称为元。其中第 i 行第 j 列的元素可以记作 A_{ij}。

特殊矩阵

1. 行数与列数都等于 n 的矩阵称为 n **阶矩阵**或 n **阶方阵**。大部分 OI 问题研究矩阵的都是方阵。
2. 只有主对角线（所有 a_{ii}）为 1，其余位置为 0 的方阵称为**单位矩阵**，一般记为 E 或 E_n（也有记作 I）。
3. 只有主对角线上有非零值，其余位置均为 0 的方阵称为**对角矩阵**。
4. 如果方阵主对角线下方均为 0，称为**上三角矩阵**；如果方阵主对角线上方均为 0，称为**下三角矩阵**。
5. 只有一行的矩阵称为**行矩阵**或**行向量**。
6. 只有一列的矩阵称为**列矩阵**或**列向量**。

矩阵运算

对于矩阵的加法、减法和数乘，都是简单的对位运算。
矩阵乘法只有在第一个矩阵的列数等于第二个矩阵的行数的情况下才有意义。
对于一个 $m \times n$ 矩阵 A 和一个 $n \times l$ 矩阵 B，记 C 为矩阵 A 与 B 的乘积，则有 C 是一个 $m \times l$ 矩阵，且:

$$C_{ij} = \sum_{k=1}^{n} A_{ik} B_{kj}$$

矩阵乘法满足结合律和对矩阵加法的分配律，**不满足交换律。**

矩阵的行列式

由 n 阶方阵 A 的元素构成的行列式，称为 A 的行列式，记作 $\det A$ 或者 $|A|$，也可写作：

$$\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix}$$

它的值等于 $\sum\limits_{p}(-1)^{t(p)}a_{1p_1}a_{2p_2}\dots a_{np_n}$，其中 p 枚举所有长度为 n 的排列，$t(p)$ 表示排列 p 的奇偶性，也就是排列 p 的逆序对数量的奇偶性。

行列式求值

行列式有如下几条性质：

1. 对换行列式的两行（列），行列式变号。
2. 行列式的某一行（列）中的所有元素都乘同一个数 k，等于用数 k 乘此行列式。
3. 把行列式的某一行（列）的各元素乘同一数然后加到另一行（列）对应元素上去，行列式不变。

事实上这三个性质也就对应了矩阵的三种初等行（列）变换。

同时，对于一个上三角矩阵 A 的行列式，由于主对角线下方都是 0，因此它的值就等于 $\prod_{i=1}^{n} a_{ii}$。因此，如果能够通过上面的三种操作，将普通的行列式变成上三角行列式，就可以直接求值。

行列式求值

可以通过高斯消元法来实现这一点：
找到第一列非零的一行（如果不存在说明行列式的值为 0），通过 1 性质将其与第一行交换，并通过 2 性质将第一项的系数变为 1。然后通过 3 性质将其余行的第一列都变为 0。
此时第一列只有第一行非零，已经符合上三角矩阵的结构，暂时将第一行和第一列除去，递归处理规模为 $n-1$ 的子问题。总时间复杂度为 $O(n^3)$。

矩阵的逆

对于 n 阶方阵 A，如果有一个 n 阶方阵 B 满足：$AB = BA = E$。则说明矩阵 A 是可逆的，并把矩阵 B 称为矩阵 A 的逆矩阵，记作 A^{-1}。
如果 A 可逆，则 A 的逆矩阵是唯一的，且 $|A| \neq 0$。

求解逆矩阵

如果矩阵 A 可逆，则可以通过初等行变换来是的其变成单位矩阵 E。而初等行变换的作用相当于矩阵，也就意味着上文操作中若干初等行变换的作用等价于乘上 A^{-1}。

则对单位矩阵 E 进行同样的初等行变换就可以将矩阵变为 A^{-1}。

使用高斯消元法实现这一个过程：同时维护两个 n 阶矩阵，其中第一个矩阵的初值为 A，第二个矩阵的初值为 E。使用高斯消元法将第一个矩阵消成 E，同时对第二个矩阵进行相同的初等行变换，最终第二个矩阵的值即为 A^{-1}。

线性基

称一个 n 维向量组 $\{\mathbf{a}_i\}$ 是**线性无关**的，当且仅当不存在不全为零的一组数 c_i，使得 $\sum c_i\mathbf{a}_i = \mathbf{0}$。
而线性基认为是一个 n 维向量集合中极大的**线性无关**向量子集。可以证明任何向量集合存在线性基，且一个向量集合的任意线性基大小相同。

线性基构造

显然，线性基最多只会有 n 个元素。维护一个向量数组 $\mathbf{a}_i, i = 1, 2 \dots n$ 来表示最终得到的线性基。

依次加入每一个向量 $\mathbf{v}$，从高到低扫描每一位。如果 $\mathbf{v}$ 的第 x 位非零，那么就检查 $\mathbf{a}_i$：

1. 如果 $\mathbf{a}_i$ 不存在，那么令 $\mathbf{a}_i \leftarrow \mathbf{v}$，退出循环。
2. 如果 $\mathbf{a}_i$ 存在，那么令 $\mathbf{v} \leftarrow \mathbf{v} - \dfrac{v_x}{\mathbf{a}_{i,x}}\mathbf{a}_i$，继续循环下一位。

上述算法的时间复杂度为 $O(n^2m)$，其中 m 为加入的向量数量。

同时，如果一个向量可以通过上面的过程循环到最后一位，最终变为的零向量 $\mathbf{0}$，说明该向量可以被这一组线性基表示。

异或线性基

一般情况下，相较于 n 维实线性空间 $\mathbb{R}^n$ 下的**实数线性基**，OI 中研究的更多的是 n 维布尔域线性空间 $\mathbf{Z}_2^n$ 下的**异或线性基。**

在布尔域线性空间中，一个向量等价于一个 $[0, 2^n)$ 内的整数，加法等价于按位异或，数乘等价于且。

在 n 维布尔域线性空间下，一个数能够被线性基表示，等价于可以被表示成数集中若干个数的异或和，也就是数集的子集异或和。

异或线性基

利用异或线性基，能够解决如下的问题：

1. 检验一个数是否能表示成某个数集的子集异或和，以及方案数。
2. 一个数集能够表示的子集异或和的数量/第 k 大值。

异或线性基构造

构造的原理基本上实数线性基的构造是一致的：

依次加入每一个向量 $\mathbf{v}$，从高到低扫描每一位。如果 $\mathbf{v}$ 的第 x 位非零，那么就检查 $\mathbf{a}_i$：

1. 如果 $\mathbf{a}_i$ 不存在，那么令 $\mathbf{a}_i \leftarrow \mathbf{v}$，退出循环。
2. 如果 $\mathbf{a}_i$ 存在，那么令 $\mathbf{v} \leftarrow \mathbf{v} \operatorname{xor} \mathbf{a}_i$，继续循环下一位。

上面的 xor 操作可以通过位运算优化，因此上述算法的时间复杂度为 $O(\frac{n^2 m}{\omega})$，其中 m 为加入的向量数量。在 $n \le 64$ 的时候一般认为时间复杂度为 $O(nm)$。

LGV 引理

行列式在 OI 中的应用较为广泛，其中比较经典的一类就是利用 LGV 引理来解决图上不交路径计数的问题。

LGV 引理

对于一张有边权的**有向无环图** G，定义一条路径 P 的权值 $w(P)$ 为路径上所有边的边权之积。
定义 $e(u,v)$ 为 u 到 v 的**所有**路径 P 的权值之和，也就是 $\sum\limits_{P:u\to v} w(P)$。
对于两个大小为 n 的 G 的点集的子集 A 和 B，分别称其为起点集合和终点集合，则一组从 A 到 B 的不交路径 S 为：S_i 是一条从 A_i 到 $B_{\sigma(S)_i}$ 的路径（其中 $\sigma(S)$ 是一个与 S 对应的排列），对于任何 $i \neq j$，路径 S_i 和 S_j 不存在公共点。

LGV 引理

那么 LGV 引理说的是，对于矩阵：

$$M = \begin{bmatrix} e(A_1, B_1) & e(A_1, B_2) & \dots & e(A_1, B_n) \\ e(A_2, B_1) & e(A_2, B_2) & \dots & e(A_2, B_n) \\ \vdots & \vdots & & \vdots \\ e(A_n, B_1) & e(A_n, B_2) & \dots & e(A_n, B_n) \end{bmatrix}$$

有 $\det M = \sum_{S:A\to B} (-1)^{t(\sigma(S))} \prod_{i=1}^{n} w(S_i)$。

LGV 引理证明

现根据行列式的定义来求解 $\det M$，则有：

$$
\begin{aligned}
\det M &= \sum_{p}(-1)^{t(p)}\prod_{i=1}^{n} e(A_i, B_{p_i}) \\
&= \sum_{p}(-1)^{t(p)}\prod_{i=1}^{n}\Big(\sum_{P_i : A_i \to B_{p_i}} w(P_i)\Big) \\
&= \sum_{p}(-1)^{t(p)}\sum_{P: A\to B \wedge \sigma(P)=p}\prod_{i=1}^{n} w(P_i) \\
&= \sum_{P: A\to B}(-1)^{t(\sigma(p))}\prod_{i=1}^{n} w(P_i)
\end{aligned}
$$

因此 $\det M$ 就是对从 A 到 B 的任意路径组（没有不交限制）的带符号求和。

LGV 引理证明

对于一个存在相交的路径组：P。我们找到其中 i 最小的那一条和其他路径有交的路径 P_i，并找到其路径上的第一交点 u，假设其与 P_j 相交。
可以找到这样一个路径组 P'：对于 $k \neq i \wedge k \neq j$，$P'_k = P_k$；记 $P_i : a_i \to u \to b_{\sigma(P)_i}$，$P_j : a_j \to u \to b_{\sigma(P)_j}$，则令 $P'_i : a_i \to u \to b_{\sigma(P)_j}$，$P'_j : a_j \to u \to b_{\sigma(P)_i}$。
此时，有 $\prod_{i=1}^{n} w(P_i) = \prod_{i=1}^{n} w(P'_i)$，但交换了 P_i 和 P_j 的终点，$\sigma(P)$ 和 $\sigma(P')$ 的奇偶性不同，因此 $(-1)^{t(\sigma(P))} \prod_{i=1}^{n} w(P_i)$ 和 $(-1)^{t(\sigma(P'))} \prod_{i=1}^{n} w(P'_i)$ 互为相反数，在求和中将会抵消。
对于每一个有交的路径，其对于 $\det M$ 的贡献都会被抵消掉，因此剩余的就是不交路径组，也就是 LGV 引理中的右式了。

LGV 引理应用

由于是对于不交路径组的带符号求和，所以 LGV 引理难以直接统计所有不交路径组的权值和。因此，在实际问题中，有如下几种解决方案：

1. 题意要求统计的就是带符号和。
2. 题目中的图为特殊图，使得不交路径的起点和终点对应是固定的，只存在一种或奇偶性相同的几种 $\sigma(S)$。
3. 只需要检验不交路径组的存在性，给边随机赋权，查询 $\det M$ 是否非零。

斐波那契数列

简要题意

已知斐波那契数列的定义为 $F_0 = 0$，$F_1 = 1$，$F_n = F_{n-1} + F_{n-2}(n \geq 2)$。求 $F_n \bmod 998244353$ 的值。

$1 \leq n < 2^{63}$。

斐波那契数列

发现 F_n 的求值是可以递推得到的，但是由于 n 过大导致复杂度无法接受，所以要尝试加速这个过程。

由于每一次递推的方式是相同的，所以如果能够讲其写成矩阵的形式，那么就可以利用矩阵快速幂来进行加速。

根据定义可知：$\begin{bmatrix} F_n \\ F_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} F_{n-1} \\ F_{n-2} \end{bmatrix}$。

记 $A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$，那么就有 $\begin{bmatrix} F_n \\ F_{n-1} \end{bmatrix} = A^{n-1} \begin{bmatrix} F_1 \\ F_0 \end{bmatrix}$，需要求解 F_n 也就是要求解 A^{n-1}。

总时间复杂度 $O(\log n)$。

NOI2012 随机数生成器

简要题意

有一个序列 $\{X_n\}$，满足 $X_{n+1} = (aX_n + c) \bmod m$，求 $X_n \bmod g$ 的值。
$1 \le n, m, a, c, X_0 \le 10^{18}$，$1 \le g \le 10^8$。

NOI2012 随机数生成器

对于线性递推式，可以将转移写成矩阵的形式：

$$\begin{bmatrix} X_{n+1} \\ 1 \end{bmatrix} = \begin{bmatrix} a & c \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_n \\ 1 \end{bmatrix}$$

记 $A = \begin{bmatrix} a & c \\ 0 & 1 \end{bmatrix}$，那么就有 $\begin{bmatrix} X_n \\ 1 \end{bmatrix} = A^n \begin{bmatrix} X_0 \\ 1 \end{bmatrix}$，可以通过矩阵快速幂来求解 A^n。

总时间复杂度 $O(\log n)$。

NOI2013 矩阵游戏

简要题意

按照如下方式生成一个 $n \times m$ 的矩阵 F：

- $F_{1,1} = 1$
- $F_{i,j} = a \times F_{i,j-1} + b$，$j \neq 1$。
- $F_{i,1} = c \times F_{i-1,m} + d$，$i \neq 1$。

问 $F_{n,m} \bmod (10^9 + 7)$ 的值是多少。

$1 \leq n, m \leq 10^{10^6}$，$1 \leq a, b, c, d \leq 10^9$。

NOI2013 矩阵游戏

与上一题类似，对于行内和行间的转移均可以写成矩阵的形式：

$$\begin{bmatrix} F_{i,j} \\ 1 \end{bmatrix} = \begin{bmatrix} a & b \\ 0 & 1 \end{bmatrix} \begin{bmatrix} F_{i,j-1} \\ 1 \end{bmatrix}, \begin{bmatrix} F_{i,1} \\ 1 \end{bmatrix} = \begin{bmatrix} c & d \\ 0 & 1 \end{bmatrix} \begin{bmatrix} F_{i-1,m} \\ 1 \end{bmatrix}$$

NOI2013 矩阵游戏

记 $A = \begin{bmatrix} a & b \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} c & d \\ 0 & 1 \end{bmatrix}$。

那么就有：

$$\begin{aligned}\begin{bmatrix} F_{i,1} \\ 1 \end{bmatrix} &= B \begin{bmatrix} F_{i-1,m} \\ 1 \end{bmatrix} \\ &= BA^{m-1} \begin{bmatrix} F_{i-1,1} \\ 1 \end{bmatrix} \\ &= (BA^{m-1})^{i-1} \begin{bmatrix} F_{1,1} \\ 1 \end{bmatrix}\end{aligned}$$

NOI2013 矩阵游戏

因此，想要求解 $F_{n,m}$，就是要求解矩阵 $A^{m-1}(BA^{m-1})^{n-1}$ 的值。
先利用矩阵快速幂求解 A^{m-1}，然后求解 $(BA^{m-1})^{n-1}$ 即可。
时间复杂度 $O(\log n + \log m)$。

CQOI2013 新 Nim 游戏

简要题意

现在有 n 堆石子，其中第 i 堆有 a_i 个石子，现在已经游戏如下：

1. 在第一个回合中，每一个人可以取走**任意堆**石子（可以不去，但不能全部取完）。
2. 从第二回合开始，每一个每次选择一堆石子，从中取走至少一颗石子（可以全部取完），取走最后一颗石子的人获胜。

问先手是否存在必胜策略。如果有，他第一回合至少要取走多少颗石子。

$1 \leq n \leq 300$，$1 \leq a_i \leq 10^9$。

CQOI2013 新 Nim 游戏

不考虑第一回合，那么先手有必胜策略当且仅当所有的 a_i 异或和不为 0。
考虑第一回合后手的操作，假设其剩余的石子集合为 S，那么其可以保留该石子集合的任意子集。而后手只需要是的保留的子集异或和为 0，他就可以获得胜利。
因此，在第一回合先手操作后，剩余的 S 集合不能存在一个非空子集，使得其中所有元素的异或和为 0。

CQOI2013 新 Nim 游戏

这也就是要求 S 中的元素线性无关，而同时由于先手第一回合要尽可能少地取石子，那么也就要让 S 中的元素和尽可能大。
因此可以得到，S 应当是元素和最大的那一组线性基。考虑构造线性基的过程，每一次是能加入线性基则贪心地加入，因此可以按照 a_i 的大小从大到小尝试加入，最终得到的线性基就是 $\sum a_i$ 最大的。
时间复杂度 $O(k \log V)$。

CF895C Square Subsets

简要题意

有 n 个数 a_i，问有多少种选择至少一个数的方法，使得选出的数的乘积是完全平方数。

$1 \le n \le 10^5$，$1 \le a_i \le 70$。

CF895C Square Subsets

只考虑最终的乘积是否是完全平方数，因此可以将 a_i 中的所有的平方因子除去。

同时不同的质数之间数独立的，那么对于某一个质数 p。在去掉所有的平方因子之后，只剩下 p^0 和 p^1 两种可能。如果在运算的过程之中仍然保持去除所有的平方因子，那么就是对质数的指数做异或操作。

≤ 70 的质数一共有 19 个，因此每一个数对于一个 19 维的布尔向量。而题目所有就是有多少种不同的线性组合方式，使得得到的和为 $\mathbf{0}$。

CF895C Square Subsets

先建出着 n 个数对应的异或线性基，那么不在线性基内的所有元素，它们任意的线性组合得到的结果，都可以**唯一**对应一个线性基内元素的线性组合，使得最终的和为 **0**。
假设线性基内元素的数量为 k，则最终答案不在线性基内元素任选的方案数，即为 $2^{n-k}-1$，其中的 -1 是要去掉所有元素都不选择的那组方案。

CF1163E Magical Permutation

简要题意

给定一个大小为 n 的正整数集合 S，要求找到最大的 x，满足：存在一个 0 到 $2^x - 1$ 构成的排列，使得排列中任意相邻的两项的异或和都是 S 中的元素。并给出构造。

$1 \leq n, S_i \leq 2 \times 10^5$。

CF1163E Magical Permutation

显然 $x \le \left\lfloor \log_2 \max_{a \in S} a \right\rfloor + 1$，因此问题可以变成检验每个 x 是否满足条件。
假设 $p_0, p_1 \dots p_{2^x-1}$ 是一个满足条件的排列，不妨假设 $p_0 = 0$。否则令 $p_i \leftarrow p_i \operatorname{xor} p_0$，新的排列有 $p_0 = 0$ 且仍然满足条件。
记 $b_i = p_{i-1} \operatorname{xor} p_i$，因为有 $b_i < 2^x$，所以可以只保留 S 中 $< 2^x$ 的元素。
同时，有 $p_i = b_1 \operatorname{xor} b_2 \operatorname{xor} \dots b_i$，又因为 $b_i \in S$。所以，就是 p_i 等于 S 的某一个子集异或和，或者说，可以被 S 中的元素线性表示。

CF1163E Magical Permutation

又因为 $p_0, p_1 \dots p_{2^x-1}$ 是 0 到 2^x-1 的一个排列，因此 S 中的元素可以线性表示 $0 \sim 2^x-1$ 中的所有数。

这就说明 S 的线性基的大小为 x，这一点可以直接通过构造线性基来检验。

CF1163E Magical Permutation

假设 S 的一组线性基为 $t_1, t_2, \dots t_x$，那么这 x 个数是否选择的共 2^x 种方案 (通过状压是否选择也对应 0 到 2^x-1 内的一个数）得到的异或和与 0 到 2^x-1 的数一一对应。

如果我们能够找到一个排列 $q_0=0, q_1, q_2 \dots q_{2^x-1}$，使得相邻两项的二进制表示只有一位不同，那么就可以通过上面一一对应的方式生成题目要求的排列 p。

而排列 q 序列的定义就是**格雷码**，下面给出格雷码的一种构造方式：

记 lowbit(x) 表示 x 二进制表示下最小的非零位对应的 2 的次幂，例如 lowbit$(1)=1$，lowbit$(6)=2$。令 $q_i=q_{i-1}$ xor lowbit(i)，即可得到一组格雷码。

CF348D Turtles

题目大意

给定一个 $n \times m$ 的网格，某些格子上有障碍物。有两只乌龟（不区分）想从 $(1,1)$ 走到 (n,m)，两只乌龟每一次只能向右或者向下，且除起点终点外不会经过同一个格子的方案数有多少种。

两种方案不同当且仅当存在至少一个格子在一个方案中被经过，另一个方案中未被经过。

$1 \leq n,m \leq 3000$。

CF348D Turtles

这两只乌龟必然是一只从 $(1,2)$ 出发走到 $(n-1,m)$，另一只从 $(2,1)$ 出发走到 $(n,m-1)$，且路径不能相交。

使用 LGV 引理，分别记从 $(1,2)$ 走到 $(n-1,m)$，$(n,m-1)$ 的路径方案数为 a，b；从 $(2,1)$ 走到 $(n-1,m)$，$(n,m-1)$ 的路径方案数为 c，d。

那么从这起点走到终点的路径的带符号和为 $\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$。但由于不存在从 $(1,2)$ 走到 $(n,m-1)$ 和 $(2,1)$ 走到 $(n-1,m)$ 且路径不交的方案。因此该行列式的值 $ad-bc$ 就是题目所求。

a,b,c,d 的具体值可以通过 DP 计算，时间复杂度 $O(nm)$。

P6657 【模板】LGV 引理

简要题意

有一个 $n \times n$ 的棋盘，左下角为 $(1,1)$，右上角为 (n,n)，若一个棋子在点 (x,y)，那么走一步只能走到 $(x+1,y)$ 或 $(x,y+1)$。

现在有 m 个棋子，第 i 个棋子一开始放在 $(a_i,1)$，最终要走到 (b_i,n)。问有多少种方案，使得每个棋子都能从起点走到终点，且对于所有棋子，走过路径上的点互不相交。输出方案数 $\bmod 998244353$ 的值。

两种方案不同当且仅当存在至少一个棋子所经过的点不同。

$1 \le n \le 10^6$，$1 \le m \le 100$，$1 \le a_1 \le a_2 \dots \le a_m \le n$，$1 \le b_1 \le b_2 \dots \le b_m \le n$。

P6657 【模板】LGV 引理

和上一题类似，发现不存在其他起点和终点匹配的方法使得存在不交路径组。
因此，利用 LGV 引理计算出的就是所求答案。
而从 $(a,1)$ 走到 (b,n) 的路径数，可以直接使用组合数计算，等于 $\binom{n-1+b-a}{b-a}$。
时间复杂度 $O(m^3)$。

有向图哈密顿路

题目大意

给定一张 n 个点，m 条边的有向图，要求找到一条 k 个点的路径使得路径上的点互不相同。

$1 \le n \le 100$，$1 \le m \le 200$，$1 \le k \le 15$。

有向图哈密顿路

发现核心的问题在于如何让路径上的点互不相同，需要一种在经过重复点的就一定不会统计的方法。

根据行列式的第一个性质，可以得到推论：如果行列式的两行相同，行列式的值为 0。

因此，给每一个点赋值一个 k 维向量 $\mathbf{v}_i$，那么对于一个 k 个点的路径 $a_1, a_2 \dots a_k$，如果 $\det \begin{bmatrix} \mathbf{v}_{a_1} \\ \mathbf{v}_{a_2} \\ \vdots \\ \mathbf{v}_{a_k} \end{bmatrix} = 0$，说明 $a_1, a_2 \dots a_k$ 中存在重复点；否则，不存在重复点。

如果给 $\mathbf{v}_i$ 进行随机赋权，上面的判断的错误率将会是 $\dfrac{1}{Mod}$ 的。

有向图哈密顿路

使用上页方式检验，不需要记录经过的点，只需要对其进行数值计算。而对于上面行列式的求值，可以直接根据定义进行：
记 $f_{i,j,S}$ 表示考虑了路径前 i 个点，第 i 个点为 j 的情况下，前 i 行行列式求值选择的 p_i 集合为 S 的情况下前 i 行的带符号和。
转移的时候直接枚举边，以及这一行选择的 p_i 进行计算。最终 $f_{k,v,\{1,2...k\}}$ 非零的位置 v 就是一个可行的终点，可以根据转移倒推来得到路径的构造。
时间复杂度 $O(mk^2 2^k)$。

NOI2021 密码箱

题目大意

对于一个数列 $a_0, a_1 \dots a_k$，有函数 f 如下：

$$f(a_0, a_1 \dots a_k) = \begin{cases} a_0 & , k = 0 \\ f(a_0, a_1 \dots a_{k-1} + \frac{1}{a_k}) & , k \geq 1 \end{cases}$$

根据一系列操作来生成这个数列 a，初始时有 $a_0 = 0, a_1 = 1$：

- W 操作：给数列的最后一向加 1。
- E 操作：如果数列的最后一项为 1，则给倒数第二项加 1；否则给最后一项减 1，接着在数列尾再加两项，两项的值都是 1。

由于题目的后半部分为数据结构内容与主题无关，现在只需要考虑如何使用矩阵快速维护 E 操作和 W 操作对于数列的函数值 f 的影响。

NOI2021 密码箱

直接将 f 的求值过程展开，就有 $f = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \ddots}}}$，这是一个连分数。

而 $\cfrac{1}{v + \cfrac{a}{b}} = \cfrac{b}{a + vb}$，如果将分子分母看作一个二维列向量，就有：

$$\begin{bmatrix} 0 & 1 \\ 1 & v \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} b \\ a + vb \end{bmatrix}$$

那么要维护 f 的值，也就是维护 $\begin{bmatrix} 0 & 1 \\ 1 & a_0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & a_1 \end{bmatrix} \dots \begin{bmatrix} 0 & 1 \\ 1 & a_k \end{bmatrix}$ 的值。

现在依次考虑 W 操作和 E 操作对于上面这个矩阵连乘的影响。

NOI2021 密码箱 W 操作

只和最后一项有关，因此只考虑最后一项，有：

$$\begin{bmatrix} 0 & 1 \\ 1 & a+1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & a \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

所以 W 操作等价于右乘上矩阵 $\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$。

NOI2021 密码箱 E 操作

在数列最后一项为 1 的时候，修改和后两项有关：

$$
\begin{aligned}
\begin{bmatrix} 0 & 1 \\ 1 & a+1 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 1 & a \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 1 \\ 1 & a \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 1 \\ 1 & a \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 0 & -1 \\ 1 & 2 \end{bmatrix}
\end{aligned}
$$

NOI2021 密码箱 E 操作

在数列最后一项不为 1 的时候，修改之和最后一项有关，有：

$$
\begin{aligned}
\begin{bmatrix} 0 & 1 \\ 1 & a-1 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 1 & a \end{bmatrix}\begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 1 \\ 1 & a \end{bmatrix}\begin{bmatrix} -1 & 0 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 1 \\ 1 & a \end{bmatrix}\begin{bmatrix} 0 & -1 \\ 1 & 2 \end{bmatrix}
\end{aligned}
$$

发现 E 操作的两种情况都等价于右乘上矩阵 $\begin{bmatrix} 0 & -1 \\ 1 & 2 \end{bmatrix}$。

由此，W 和 E 操作以及 f 的求值都变成了容易维护的矩阵乘法的形式。

CF1100F Ivan and Burgers

题目大意

有一个长度为 n 的序列 a_i，q 次询问，每一次询问一个区间 l, r，问 $a_l, a_{l+1}, \ldots a_r$ 中选择若干个数的异或和最大为多少。

$n, q \leq 5 \times 10^5$，$a_i \leq 10^6$。

CF1100F Ivan and Burgers

询问相当于要求区间 $[l,r]$ 内的 c_i 构成的一组线性基。
发现如果想要合并两个线性基 S_1，S_2，唯一可行的方法是将 S_2 中的每一个元素尝试加入到 S_1 中。
单次合并的时间复杂度就为 $O(\log^2 V)$，无法接受。

CF1100F Ivan and Burgers

因此要考虑从构造线性基的角度去分析：
一般构造线性基的方法是增量构造，枚举 $i = 1, 2 \dots n$，依次尝试将 a_i 加入到线性基中。
那么在加入了 a_r 之后，当前的线性基对应的就是 $[1, r]$ 构成的线性基。如果此时能够想办法仅考虑 $l \le i$ 的 a_i，那么也就得到了 $[l, r]$ 构成的线性基。
换而言之，我们希望当前的线性基中，保留的 a_i 的 i 尽可能大，加入的时间尽可能晚。

CF1100F Ivan and Burgers

在前面构造线性基的方法中，如果当前加入的是 **v**，会在 $\mathbf{a}_i$ 存在的时候对 **v** 进行修改。
但是由于 **v** 加入的时间比 $\mathbf{a}_i$ 晚，而 **v** 和 $\mathbf{a}_i$ 在当前位置的地位是等价的。
因此可以将 $\mathbf{a}_i$ 和 **v** 交换，让加入时间更晚的 **v** 留下，让原本线性基中的 $\mathbf{a}_i$ 代替 **v** 去做接下来的处理。

CF1100F Ivan and Burgers

在维护线性基的同时，维护线性基内每一个元素对应的原数下标 id_i。
在查询 $[l,r]$ 的问题是，只考虑那些 $l \le id_i$ 的元素，即可在 $O(\log V)$ 的时间复杂度内查询答案。
这样的结构，也被称为**前缀线性基**或者**时间戳线性基**。
总时间复杂度 $O((n+q)\log V)$。

PA 2021 Fiolki 2

题目大意

有一张 n 个点 m 条边的有向无环图。记 $f(l,r)$（$k < l \le r \le n$）表示以 $1 \sim k$ 中的点为起点，$l \sim r$ 中的点为终点，最多能够选出多少条路径，使得任意两条路径不存在公共节点。

对于 $x = 0, 1 \dots k$，问有多少对 l, r 满足 $f(l,r) = x$。

$n \le 10^5$，$m \le 10^6$，$k \le 50$。

PA 2021 Fiolki 2

检验是否存在不交路径，可以对边进行随机赋权，然后使用 LGV 引理来检验。
对于每一个 $k < i \leq n$ 维护 $1 \sim k$ 到 i 的所有路径的权值和，将其看作一个 k 维向量。那么如果 $f(l,r) = k$，根据 LGV 引理，就是能够找到 $a_1, a_1 \dots a_k \in [l,r]$，使得这 k 个点对应的向量排成一列构成的矩阵的行列式的值不为 0。
也就是要能够在 $[l,r]$ 内找到 k 个线性无关的向量。

PA 2021 Fiolki 2

依此类推，有 $f(l,r) \geq x$，就意味着能够在 $[l,r]$ 内找到 x 个线性无关的向量。
所以 $f(l,r)$ 就是 $[l,r]$ 的每一个点对应的向量构成的线性基的大小。
对于确定的 r，$f(l,r)$ 的值随着 l 的减小而增大，而如果 $f(l,r) \neq f(l+1,r)$，也就意味着 a_l 和 $a_{l+1} \dots a_r$ 都线性无关，可以加入线性基中。需要找到所有这样的 l。
而想要找到所有这样的 l，可以直接维护时间戳线性基。
时间复杂度 $O(mk + nk^2)$。

SNCPC2024 最大流

题目大意

给定一个 n 个点 m 条边的有向无环图，每条边的容量为 1。对于每一个点 v，记 f_v 表示以 1 为源点，v 为汇点的最大流大小。

对于 $i = 2 \sim n$，求 $\min(f_i, k)$ 的值。

$n \le 10^5$，$m \le 2 \times 10^5$，$k \le 50$。

SNCPC2024 最大流

相当于问从 1 到 v 最多有多少条**边不交**的路径，并且对 k 取 min。添加一个点 0，令其向 1 连接 k 条边，即可将对 k 取 min 的限制去掉。

由于 LGV 引理检验的的点不交，所以考虑重新建图：每一条边对应一个节点，对于一个点的所有入边向出边连接带有随机权值的边。

i 点的答案就等于 i 的所有入边对应向量构成的线性基大小即可。

SNCPC2024 最大流

但是如果存在一个点，它的入边和出边都很多，就会导致直接建边的复杂度变劣，考虑对其进行优化。
发现所有入边向所有出边连边，并对每条边随机赋权，就是让出边的向量等于入边向量随机线性组合。
因此，这样的操作等效于：只保留入边的一组线性基，令每条出边的权值变成线性基内向量的随机线性组合。
总时间复杂度 $O((n+m)k^2)$。

Q&A

谢谢大家。