浅谈非确定性算法

zhoukangyang

学军中学

January 18th, 2025

引言

非确定性算法是信息学竞赛中很重要的一部分，例如哈希、乱搞等做法均利用了随机化的思想。今天我将给大家讲解随机化算法在 OI 中更为复杂的运用。

CF1746F

Problem

维护一个序列，要求支持单点修改和查询一个区间是否满足所有元素出现次数都是 k 的倍数（k 每次给定）。

$n, q \leq 3 \times 10^5$

CF1746F

考虑对于每个值，将其随机替换为 $0/1$。查询时计算区间内所有元素的和并判断其是否是 k 的倍数。

CF1746F

考虑对于每个值，将其随机替换为 $0/1$。查询时计算区间内所有元素的和并判断其是否是 k 的倍数。
该做法正确率至少为 $1/2$。因此做 40 次即可保证正确率。

NOI 2013 向量内积

Problem

给定 n 个 d 维向量 a_i。
问是否存在两个向量的内积是 k 的倍数。即找一对 (i, j) 使得 $\sum_k a_{i,k}a_{j,k} = 0$。
$n \leq 10^5, d \leq 30, k \in \{2, 3\}$。

NOI 2013 向量内积

首先考虑 $k = 2$ 怎么做。
如果不存在这样的 (i, j)，那么必然有 $\sum_k a_{i,k} a_{j,k} = 1$。

NOI 2013 向量内积

首先考虑 $k = 2$ 怎么做。
如果不存在这样的 (i, j)，那么必然有 $\sum_k a_{i,k} a_{j,k} = 1$。
考虑随机化，随机一个集合 S，对所有 i 计算 i 和 S 的总和的内积。每次正确率至少为 $1/2$，做多次即可。

NOI 2013 向量内积

首先考虑 $k=2$ 怎么做。
如果不存在这样的 (i,j)，那么必然有 $\sum_k a_{i,k}a_{j,k}=1$。
考虑随机化，随机一个集合 S，对所有 i 计算 i 和 S 的总和的内积。每次正确率至少为 $1/2$，做多次即可。
如果做 T 次，复杂度是 $\mathcal{O}(\frac{Tnd}{\omega})$，错误率 2^{-T}。

NOI 2013 向量内积

$k=3$ 可以考虑计算内积的平方：如果内积非 0，那么内积的平方必然为 1。
$(\sum_{k=1}^{d} a_{i,k}a_{j,k})^2 = \sum_{x,y} a_{i,x}a_{i,y}a_{j,x}a_{j,y}$，可以转化为长度为 k^2 的向量的内积。
接下来就能进行相同操作了，复杂度 $\mathcal{O}(\frac{Tnd^2}{\omega})$，错误率 2^{-T}。

原创题

给定一个序列，要求支持：

- 区间翻转。
- 查询从区间中选出一个子集能获得的最大异或和。

$n, q \leq 5 \times 10^4, 0 \leq a_i < 2^{60}$。

原创题-题解

考虑维护 100 个序列，每个序列的第 i 个元素有 $1/2$ 的概率为 0 有 $1/2$ 的概率为 a_i。
做查询时，只需对每个序列都求出区间异或和，然后假装这些元素形成的线性基是原区间的线性基即可。

原创题-正确性分析

显然我们得到的线性基应是真正线性基的子空间。所以只需证明其有高概率是真正的线性基即可。

原创题-正确性分析

显然我们得到的线性基应是真正线性基的子空间。所以只需证明其有高概率是真正的线性基即可。
首先我们可以将证明归约至区间线性基恰好能表示 $[0, 2^l - 1]$ 的情况。

原创题-正确性分析

显然我们得到的线性基应是真正线性基的子空间。所以只需证明其有高概率是真正的线性基即可。

首先我们可以将证明归约至区间线性基恰好能表示 $[0, 2^l - 1]$ 的情况。

如果我们的线性基不完整，那么一定存在一个 l 维向量 v 使得 v 和线性基中的所有元素点乘均为 0。这对于一个序列来说概率是 $\frac{1}{2}$，100 个都满足就是 2^{-100}。

原创题-正确性分析

显然我们得到的线性基应是真正线性基的子空间。所以只需证明其有高概率是真正的线性基即可。
首先我们可以将证明归约至区间线性基恰好能表示 $[0, 2^l - 1]$ 的情况。
如果我们的线性基不完整，那么一定存在一个 l 维向量 v 使得 v 和线性基中的所有元素点乘均为 0。这对于一个序列来说概率是 $\frac{1}{2}$，100 个都满足就是 2^{-100}。
因此如果想要在一组询问中有高概率获得正确答案，那么需要 $\mathcal{O}(\log V)$ 个序列。q 组询问就是 $\mathcal{O}(\log qV)$ 个。
时间复杂度 $\mathcal{O}((n + q \log nV) \log qV)$。

THUPC2019 找树

Problem

定义 $\otimes_1, \otimes_2, \otimes_3$ 分别为按位与、按位或、按位异或运算。记 a_i 表示 a 的从低位到高位的第 i 个二进制位。定义一个作用在 w 位二进制数上的新运算 $\oplus$，满足对于结果 $a \oplus b$ 的每一位 $(a \oplus b)_i$ 有 $(a \oplus b)_i = a_i \otimes_{o_i} b_i$。不难验证 $\oplus$ 运算满足结合律和交换律。
给出一张 n 个点 m 条边的无向图，每一条边的权值是一个 w 位二进制数（即小于 2^w 的非负整数）。请你找一棵原图的生成树。设你找出的生成树中的边边权分别为 $v_1, \cdots, v_{n-1}$，请你最大化 $v_1 \oplus v_2 \oplus \cdots \oplus v_{n-1}$。
$n \leq 70, m \leq 5000, 1 \leq w \leq 12$。

THUPC2019 找树

不考虑最优化了，考虑用 FWT+ 矩阵树定理计数！
方案数可能很大，对大质数取模。
时间复杂度 $\mathcal{O}(n^3 2^w)$。

一般图最大匹配

给定一张图无向图，求其最大匹配。
$n \leq 500$

一般图最大匹配

考虑设计矩阵 G:
对于 $i \leq j$。如果 i 到 j 有边，那么 $G_{i,j} = x_{i,j}, G_{j,i} = -x_{i,j}$。否则 $G_{i,j} = G_{j,i} = 0$。
结论：$\det(G) \neq 0$ 等价于图存在完美匹配。

一般图最大匹配

证明可以考虑 $\det(G)$ 的定义：$\sum_p sgn(p) G_{i,p_i}$。如果该式存在贡献，那么 i 到 p_i 必然有边。产生贡献的边形成了若干个环。
注意到如果这些环中有奇环，那么把环翻转后 $sgn(p)$ 恰好取反，因此奇环的两种情况抵消。
所以只需要考虑只有偶环的情况。此时每个偶环内的点必然存在匹配。而如果存在匹配，那么这些点两两成环必然合法。
所以 $\det(G) \neq 0$ 和存在完美匹配等价。

一般图最大匹配

考虑怎么求最大匹配：合理猜测，答案是 $\frac{rank(G)}{2}$。
首先不难发现，取出最大匹配内点对应的行，这些行线性无关，说明该矩阵秩大于等于 $2 \times maxmatch$。
另一方面，考虑如果存在一个 $r \times r$ 的子矩阵使得其 $\det \neq 0$，那么观察其对应的所有边。可以利用这些边说明 $2 \times maxmatch \geq r$。

一般图最大匹配

考虑怎么求最大匹配：合理猜测，答案是 $\frac{rank(G)}{2}$。
首先不难发现，取出最大匹配内点对应的行，这些行线性无关，说明该矩阵秩大于等于 $2 \times maxmatch$。
另一方面，考虑如果存在一个 $r \times r$ 的子矩阵使得其 $\det \neq 0$，那么观察其对应的所有边。可以利用这些边说明
$2 \times maxmatch \geq r$。
随机代值求 rank 即可。
时间复杂度 $\mathcal{O}(n^3)$。

QOJ8830

Problem

给定一个 $n \times n$ 的矩阵 a，和一个质数 k。
对于每个 $0 \le r < k$，问是否存在一个排列 p 使得 $(\sum_{i=1}^{n} a_{i,p_i}) \bmod k = r$。
原题：保证 $k = 5$，$n \le 1000$。
加强：保证 $k \le 30, n \le 3000$。

QOJ8830-原题做法 1

什么时候有无论如何选择排列总和都是唯一的？

QOJ8830-原题做法 1

什么时候有无论如何选择排列总和都是唯一的？
这个时候必然有 $\forall i_1, j_1, i_2, j_2, a_{i_1,j_1} + a_{i_2,j_2} = a_{i_1,j_2} + a_{i_2,j_1}$。同时
这条性质还有一个推论：存在 x, y 序列使得 $a_{i,j} = x_i + y_j$。

QOJ8830-原题做法 1

什么时候有无论如何选择排列总和都是唯一的？
这个时候必然有 $\forall i_1, j_1, i_2, j_2, a_{i_1,j_1} + a_{i_2,j_2} = a_{i_1,j_2} + a_{i_2,j_1}$。同时这条性质还有一个推论：存在 x, y 序列使得 $a_{i,j} = x_i + y_j$。
此时我们可以通过给一整行/一整列加一个数把整个矩阵清零。

QOJ8830-原题做法 1

什么时候有无论如何选择排列总和都是唯一的？
这个时候必然有 $\forall i_1, j_1, i_2, j_2, a_{i_1,j_1} + a_{i_2,j_2} = a_{i_1,j_2} + a_{i_2,j_1}$。同时这条性质还有一个推论：存在 x, y 序列使得 $a_{i,j} = x_i + y_j$。
此时我们可以通过给一整行/一整列加一个数把整个矩阵清零。
因此只要排列总和不都唯一，我们就能选出两行 i_1, j_1, i_2, j_2 满足 $\forall a_{i_1,j_1} + a_{i_2,j_2} \neq a_{i_1,j_2} + a_{i_2,j_1}$。此时我们把这两行扔到第一行和第二行，对第三行后的矩阵继续做这个操作。

QOJ8830-原题做法 1

什么时候有无论如何选择排列总和都是唯一的？
这个时候必然有 $\forall i_1, j_1, i_2, j_2, a_{i_1,j_1} + a_{i_2,j_2} = a_{i_1,j_2} + a_{i_2,j_1}$。同时这条性质还有一个推论：存在 x, y 序列使得 $a_{i,j} = x_i + y_j$。
此时我们可以通过给一整行/一整列加一个数把整个矩阵清零。
因此只要排列总和不都唯一，我们就能选出两行 i_1, j_1, i_2, j_2 满足 $\forall a_{i_1,j_1} + a_{i_2,j_2} \neq a_{i_1,j_2} + a_{i_2,j_1}$。此时我们把这两行扔到第一行和第二行，对第三行后的矩阵继续做这个操作。
如果进行超过 k 次操作则必然有解。

QOJ8830-原题做法 1

什么时候有无论如何选择排列总和都是唯一的？

这个时候必然有 $\forall i_1, j_1, i_2, j_2, a_{i_1,j_1} + a_{i_2,j_2} = a_{i_1,j_2} + a_{i_2,j_1}$。同时这条性质还有一个推论：存在 x, y 序列使得 $a_{i,j} = x_i + y_j$。

此时我们可以通过给一整行/一整列加一个数把整个矩阵清零。

因此只要排列总和不都唯一，我们就能选出两行 i_1, j_1, i_2, j_2 满足 $\forall a_{i_1,j_1} + a_{i_2,j_2} \neq a_{i_1,j_2} + a_{i_2,j_1}$。此时我们把这两行扔到第一行和第二行，对第三行后的矩阵继续做这个操作。

如果进行超过 k 次操作则必然有解。

假设进行了 d（$d < k$）次操作，我们可以对 $i, j > 2d$ 将 $a_{i,j}$ 清零。原题可以直接状压 DP。

QOJ8830-原题做法 2

考虑取一个合适的质数 mod 使得 $k-1|mod$。
考虑随机一个矩阵 b，计算矩阵 $G_{i,j}=b_{i,j}x^{a_{i,j}}$ 的 $\det$，即计算 $\sum_p sgn(p)\prod_{i=1}^{n} b_{i,p_i}x^{a_{i,p_i}} \bmod (x^k-1)$。单位根反演后发现只用求行列式。
时间复杂度是 $\mathcal{O}(kn^3)$，如果实现得较好可以通过原题。

QOJ8830-加强版做法

考虑用做法 1 的方法变换矩阵；接着使用做法 2 的思路，所求即为：

$$\begin{vmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{vmatrix} = \begin{vmatrix} I & 0 \\ 0 & B_{2,2} \end{vmatrix} \begin{vmatrix} B_{1,1} & B_{1,2} \\ B_{2,2}^{-1} B_{2,1} & I \end{vmatrix}$$
$$= \left|B_{2,2}\right| \begin{vmatrix} B_{1,1} - B_{1,2} B_{2,2}^{-1} B_{2,1} & 0 \\ B_{2,2}^{-1} B_{2,1} & I \end{vmatrix}$$

这里 $B_{2,2}$ 是系数都是随机常数的矩阵，因此我们认为其可逆，且我们只需随机其逆矩阵。
至此，我们已经得到了一个可以做到 $\mathcal{O}(n^2k^2 + k^4)$ 的做法，瓶颈在于每次计算 $B_{1,2}$ 乘随机矩阵再乘 $B_{2,1}$，这部分单次要 $\mathcal{O}(n^2k)$。

QOJ8830-加强版做法

这个等价于是：对于每个列向量 $v_1 = B_{2,1,*,j}$ 和行向量 $v_2 = B_{1,2,i,*}$，将 $v_1 v_2$ 乘一个随机数加到结果中。
不难发现，这个的结果只和这些行向量/列向量的线性基有关，所以可以利用之前题目的做法：算出这些向量的随机线性组合，然后再计算答案。
这样单次复杂度降至 $\mathcal{O}(nk^2)$，瓶颈还是计算乘随机矩阵。总复杂度 $\mathcal{O}(n^2 + nk^3)$。实际上这个乘的矩阵稀疏一点也是有正确性保证的，因此复杂度还能更低。

CF1641D

Problem

给定 n 个长度为 m 的序列，每个序列有个价值。
要求找到一对 (i,j) 使得第 i 个序列和第 j 个序列的元素集合不交且价值和最大。
$n \leq 10^5, m \leq 5$。

CF1641D-题解

把每个值映射到一个 $[1, 15]$ 内的值，然后跑高维前缀和。跑 50 次这个过程即可通过。

「THUSCH 2017」巧克力

Problem

给定 n, m 和两个 $n \times m$ 的矩阵 a, c，求一个矩阵的连通块使得这个连通块内的 c 至少出现了 k 个不同的数。在最小化连通块大小的前提下最小化连通块内 a 的中位数。

$n \times m \leq 233$

「THUSCH 2017」巧克力 - 题解

将 a 矩阵的每个数映为 $[1,k]$ 的一个数。

「THUSCH 2017」巧克力 - 题解

将 a 矩阵的每个数映为 $[1, k]$ 的一个数。
可以用斯坦纳树求最小连通块，然后二分求解第二问的答案即可。

有向图哈密顿路

Problem

给定一张有向图，要求找出一条 k 个点的路径使得路径上的所有点互不相同。
$n \leq 100, m \leq 200, k \leq 15$。

有向图哈密顿路

给每个点赋一个 k 维向量 a_i，给每个边赋一个边权，给一条路径 $v_1, v_2, \ldots., v_k$ 赋值其路径上的边权乘积乘以 $\det(a_{v_1}|a_{v_2}|a_{v_3}|\ldots|a_{v_k})$。
DP 求解以每个点为终点时路径的权值和。时间复杂度 $\mathcal{O}((n+m)k2^k)$。

无向二分图哈密顿路

Problem

给定无向二分图，要求找出一条 k 个点的路径使得路径上的所有点互不相同。
$n \leq 100, m \leq 200, k \leq 25$。

无向二分图哈密顿路

考虑无向图能给我们带来什么。

无向二分图哈密顿路

考虑无向图能给我们带来什么。
对于一条路径 v，考虑找到出现最早的 (x, y) 使得 $v_x = v_y$，我们可以考虑翻转 $v_{x+1}, v_{x+2}, \ldots, v_{y-1}$ 得到另一种方案。

无向二分图哈密顿路

考虑无向图能给我们带来什么。

对于一条路径 v，考虑找到出现最早的 (x, y) 使得 $v_x = v_y$，我们可以考虑翻转 $v_{x+1}, v_{x+2}, \ldots, v_{y-1}$ 得到另一种方案。

这启发我们在特征为 2 的域（即满足 $x + x = 0$ 的域，满足条件的域可以是 $\mathbb{F}_2$ 意义下的不可约多项式。nim 积也是一个满足条件的域）下进行操作。

无向二分图哈密顿路

考虑无向图能给我们带来什么。
对于一条路径 v，考虑找到出现最早的 (x, y) 使得 $v_x = v_y$，我们可以考虑翻转 $v_{x+1}, v_{x+2}, \ldots, v_{y-1}$ 得到另一种方案。
这启发我们在特征为 2 的域（即满足 $x + x = 0$ 的域，满足条件的域可以是 $\mathbb{F}_2$ 意义下的不可约多项式。nim 积也是一个满足条件的域）下进行操作。
但是上面的映射并不形成“对合”：a, b, a 在反转后还是自己。因此考虑在计数路径的时候不计入包含 a, b, a 的路径。

无向二分图哈密顿路

但这样操作后之前的映射也会出现问题。路径 $p, a, \ldots p, a$ 在翻转后是 $p, a, p, \ldots., a$，而这类路径已经被去除了。因此我们还要考虑 x, y 在序列中作为子串出现两次的情况。

无向二分图哈密顿路

但这样操作后之前的映射也会出现问题。路径 $p, a, ...p, a$ 在翻转后是 $p, a, p, ..., a$，而这类路径已经被去除了。因此我们还要考虑 x, y 在序列中作为子串出现两次的情况。
注意到此时 x, y 必恰有一个在二分图的左侧，因此我们只需要保证二分图一侧的元素不重即可。因此只对左边的部分算行列式即可。

无向二分图哈密顿路

但这样操作后之前的映射也会出现问题。路径 $p, a, \dots p, a$ 在翻转后是 $p, a, p, \dots, a$，而这类路径已经被去除了。因此我们还要考虑 x, y 在序列中作为子串出现两次的情况。
注意到此时 x, y 必恰有一个在二分图的左侧，因此我们只需要保证二分图一侧的元素不重即可。因此只对左边的部分算行列式即可。
时间复杂度 $\mathcal{O}(2^{k/2} poly(n, k))$。

无向图哈密顿路

Problem

给定无向图，要求找出一条 k 个点的路径使得路径上的所有点互不相同。
$n \leq 100, m \leq 200, k \leq 19$。

无向图哈密顿路

考虑提前将点随机划分成两半，我们首先是要保证左部点互不相同；其次，这里还要保证右边内部不会出现两次 $x-y$。因此我们还要保证右边的边互不相同。

无向图哈密顿路

考虑提前将点随机划分成两半，我们首先是要保证左部点互不相同；其次，这里还要保证右边内部不会出现两次 $x-y$。因此我们还要保证右边的边互不相同。

随机把每个点分到左部/右部，期望能有 $1/4$ 的左-> 右边，时间复杂度 $\mathcal{O}(2^{3k/4}poly(n,k))$。

LOJ 178

Problem

给定一个 n 次多项式 f 和奇质数 p，求 f 在 $\mathbb{F}_p$ 意义下的所有根。$n \leq 100$。

LOJ 178

只保留根：求 $\gcd(f, \prod_{i=0}^{p-1}(x-i))$，即 $\gcd(f, x^p - x)$。

LOJ 178

只保留根：求 $\gcd(f,\prod_{i=0}^{p-1}(x-i))$，即 $\gcd(f,x^p-x)$。
找根：随一个多项式 h，不断求 $\gcd(f,h^{(p-1)/2}-1)$ 并往两边递归。

LOJ 178

只保留根：求 $\gcd(f, \prod_{i=0}^{p-1}(x-i))$，即 $\gcd(f, x^p - x)$。
找根：随一个多项式 h，不断求 $\gcd(f, h^{(p-1)/2} - 1)$ 并往两边递归。
时间复杂度 $\mathcal{O}(n \log^2 n \log p)$。

QOJ 4920

给定 n 和 $\varphi(n)$，要求质因数分解 n。
$n \le 2^{1500}$

QOJ 4920

首先，过滤掉 n 的为 2 的质因子。
注意到对于 $\gcd(x,n)=1$，$x^{\varphi(n)}=1$。因此考虑模仿上一题的做法。
随机 x 使得 $\gcd(x,n)=1$。对于一个奇质因子 p，类似上一题地，考虑找一个 x^r 使得能以 $1/2$ 的概率分离出 p：即 $x^r \bmod p=1$ 的概率为 $1/2$。
不难发现，$x^r=1$ 的概率为 $\frac{\gcd(r,p-1)}{p-1}$，所以我们希望 $\gcd(r,p-1)=(p-1)/2$。

QOJ 4920

观察 $x^{\varphi(n)}, x^{\varphi(n)/2}, ..., x^{\varphi(n)/2^k}$。其中 $\varphi(n)/2^k$ 是奇数。其中恰有一个满足条件。

QOJ 4920

观察 $x^{\varphi(n)}, x^{\varphi(n)/2}, \dots, x^{\varphi(n)/2^k}$。其中 $\varphi(n)/2^k$ 是奇数。其中恰有一个满足条件。
因此，每次随机一个 x，然后依次求 $\gcd(n, x^{\varphi(n)} - 1), \gcd(n, x^{\varphi(n)/2} - 1), \dots$。这样就能将 n 的所有因此分组，然后就可以递归子问题。

QOJ 4920

观察 $x^{\varphi(n)}, x^{\varphi(n)/2}, ..., x^{\varphi(n)/2^k}$。其中 $\varphi(n)/2^k$ 是奇数。其中恰有一个满足条件。
因此，每次随机一个 x，然后依次求 $\gcd(n, x^{\varphi(n)} - 1), \gcd(n, x^{\varphi(n)/2} - 1), \ldots$。这样就能将 n 的所有因此分组，然后就可以递归子问题。
底层 p^k 的情况有可能无法分解，需要特判。
复杂度是 $\mathcal{O}(\text{polylog}(n))$。

感谢大家的聆听！
感谢 EI 的博客！