试题泛做

安徽师范大学附属中学 倪星宇

No. 1

试题编号	Codechef JULY 12
试题名称	Equivalent Suffix Tries
题目大意	T 组询问，每组给定一个字符串（长度不超过 100000）。求由小写字符组成的，其后缀树（每条边只表示一个字母，且末尾不加特殊字符）与该字符串同构的字符串个数。
算法讨论	对于后缀树，我们有以下性质 ①叶子个数取决于原串中最长的是其它后缀的前缀的后缀长度，若为 L，则有 N - L 个叶子； ②与根相连的结点个数取决于原串所含不同字母的个数； ③两个后缀的 LCA 深度取决于这两个后缀的 LCP。 因此，L、字母个数、前 N - L - 1 个后缀分别与第 N - L 个后缀的 LCP 在同构的字符串间是不变的，其中第三点使得后面的一部分字母确定。我们可以枚举满足限制的前 N - L 个后缀，使得第 N - L + 1 个后缀是它的前缀，并利用 Hash 值进行判重。最终字符串的种数即 $K \cdot C_{26}^{cnt}$，其中 K 为满足要求的后缀个数，cnt 为不同字符数。
时空复杂度	L 利用 KMP 以 O(N)求出，LCP 和 Hash 各需要 O(NlogN)的时间，故时间复杂度为 O(TNlogN)；空间复杂度 O(N)。

No. 2

试题编号	Codechef JULY 12
试题名称	Dynamic GCD
题目大意	给定一个 n ($1 \le n \le 50000$)个结点的树，结点上有权值；Q ($1 \le Q \le 50000$)组操作，或是对路径上每个点加 d 的修改、或是对路径上每个点的权值取 gcd 的询问。
算法讨论	对于多个数求 gcd，我们有： $\gcd(A_1, A_2, A_3, \dots A_k) = \gcd(A_1, A_2-A_1, A_3-A_2, \dots A_k-A_{k-1})$ 故我们直接进行树链剖分，线段树中除重链的首项外均记录差分值，线段树每个结点维护区间的 gcd。询问操作直接合并重链即可；修改操作可以转化为两个单点修改。
时空复杂度	时间复杂度 $O(Q\log^3 n)$；空间复杂度 O(n)。

No. 3

试题编号	Codechef AUG 15
试题名称	Future of draughts
题目大意	给定 T ($1 \le T \le 50$)张无向图，每张图的点数 N 不超过 50 且无重边、自环。Q($Q \le 200000$)组询问，每次给定 L、R、K($1 \le K \le 10000$)。在[L, R]的图中，先选择一个起始位置，每步选择一个非空的集合，对集合中的每一张图将当前位置移动到它的一个相邻点上。问 K 步过后所有图中均回到原点的方案种数，对 1000000007 取模。

算法讨论	根据简单的容斥可以得到： $\mathrm{Ans}=\sum_{i=0}^{k}(-1)^{k-i}\cdot C_k^i\cdot\prod_{j=l}^{r}tr[(G_j+I)^i]$ 其中 G_j 是第 j 张图的邻接矩阵，I 是单位矩阵，tr 指矩阵的迹（实际上就是 i 元回路的个数）。 矩阵部分可以先暴力处理出前 N 次方矩阵的迹，然后插值得到特征多项式。从而得到一个关于迹序列的 N 阶线性递推式；求和部分将组合数拆开后容易转化为一个卷积的形式。由于 10^9+7 不是一个 NTT 模数，所以需要取三个较大的 NTT 模数使其乘积大于 $K\mathrm{mod}^2$，分别做 NTT 后利用中国剩余定理合并得到答案。需要卡常数。
时空复杂度	时间复杂度 $O(TN^4+TNK+T^2K\log K)$；空间复杂度 $O(TN+N^2+T^2+K)$。

No. 4

试题编号	Codechef AUG 15
试题名称	Simple Queries
题目大意	给定一个含 N 个正整数的数组 A。Q 次操作，包含下列五种： 1. 令[L, R]中出现的数的集合为 S，询问 $\sum_{1\le i<j<k\le\lvert S\rvert}S_iS_jS_k \bmod (10^9+7)$； 2. 将下标为 x 的元素赋值为 y； 3. 将下标为 x 的元素删除； 4. 在下标为 z 的元素之后插入元素 y； 5. 询问[L, R]内不同数的个数。 数据范围：$1\le N, Q\le 100000$。
算法讨论	离线处理所有询问。使得插入和删除操作均变成将元素标记的修改操作，从而将对动态数组 A 的原问题转化为对静态数组 B 的问题。这需要用一棵 Splay 来维护每个操作的真实下标。 对于每种不同的数建一棵平衡树，维护其下标序列，用以方便的得到和 B[i] 同值的上一个数的下标 Pre[i]、下一个数的下标 Next[i]。那么操作 5 可以改写成询问[L, R]中 Pre[i] < L 的数的个数 P_0，记这些数的和为 P_1、二次方和为 P_2、三次方和为 P_3，那么操作 1 的答案为 $\frac{P_1^3-3P_1P_2+2P_3}{6}$。把(i, Pre[i])看做点，用树状数组套线段树来维护从而支持查询 P_i，修改的同时需要在平衡树中查询、更新。
时空复杂度	时空复杂度均为 $O((N+Q)\log^2(N+Q))$。

No. 5

试题编号	Codechef AUG 13
试题名称	Music & Lyrics
题目大意	给定 W 个模式串、N 个匹配串，求 W 个模式串在每个匹配串中匹配次数的和。 数据范围： $1\le W\le 500$；$1\le \lvert P\rvert\le 5000$，|P|表示每个模式串的长度；$1\le N\le 100$；$1\le \lvert S\rvert\le 50000$，|S|表示每个匹配串的长度。

算法讨论	AC 自动机裸题。 首先对所有模式串建立 AC 自动机，得到 fail 指针后对每个模式串都做一次匹配，对匹配到的终端结点的 cnt 加 1。那么答案即每个模式串对应终端结点在 fail 树上的子树 cnt 和。
时空复杂度	时间复杂度 O(W|P| + N|S|)；空间复杂度 O(W|P|)。

No. 6

试题编号	Codechef AUG 13
试题名称	Prime Distance On Tree
题目大意	给定一个 $N(2 \le N \le 50000)$个点的无权树。求树上任取两个不同的点，其距离为质数的概率。
算法讨论	事实上即统计树上所有长度为质数的简单路径个数，并除以 N*(N-1)/2。 我们采用树分治的方法来处理。每次对于分治得到的当前重心的各个子树，按一定顺序处理，设前缀的深度计数数组为 A、当前枚举子树的深度计数数组为 B，则 $C_i = \sum A_j \times B_{i-j}$ ，这可以使用 FFT 优化。统计所有 C 中长度为质数的项和即可。
时空复杂度	时间复杂度 $O(N\log^2 N)$；空间复杂度 O(N)。

No. 7

试题编号	Codechef JUNE 11
试题名称	Attack of the Clones
题目大意	我们称定义域在所有长度为 n 的 01 序列、值域为布尔值的函数为布尔函数。把一序列看成 X 的二进制（固定为 n 位），定义下列四种集合： 1、Z：满足 f(0) = 0 的函数的集合； 2、P：满足 f(~0) = 0 的函数的集合； 3、D：对任意 X，满足 f(X) = f(~X)的函数的集合； 4、A：为满足下列条件的函数的集合：对于某一 X、i、a、b（a、b 为 bool 值），满足 f(X & a << i) = f(X & b << i)，则 f(Y & a << i) = f(Y & b << i)（Y 为任意值）。 给出一个由这四种集合符号和交、并、补、差四种集合运算以及括号组成的表达式，求最终函数集合的元素个数。 多组询问，最多 100 组，每组最多 100 个字符。n 相同，且不超过 100。
算法讨论	我们可以把全集划分成 16 个互不相交的子集来方便支持集合运算。即：只属于 Z、只属于 P、只属于 D、只属于 A、只属于 ZP、只属于 ZD……以此类推。直接计算这 16 个子集的大小比较困难。我们可以把上述集合性质中的“只”拿掉，这样很容易利用性质直接用快速幂计算。最终做一个简单的容斥，就可以解出我们想要的 16 个子集大小。 我们再以一个 16 进制二进制来表示 ZPDA（是哪些子集的并集），那么所有的集合运算都可以写成二进制运算。因而我们方便的利用栈做表达式求值，最后结果集合中仍然存在的子集的函数个数和就是我们要求的答案。
时空复杂度	时间复杂度 O(logn + S)，S 为询问串长度和；空间复杂度 O(L)，L 为最大可能串长。

No. 8

试题编号	Codechef JUNE 11
试题名称	Minesweeper Reversed
题目大意	有一 R * C 的扫雷网格，开始时所有方块都是打开的。当点击任意一个方块时，所有在扫雷规则中可能和该方块同时打开的其它方块都会关闭。求至少点击多少次可以关闭所有方块。 数据范围：$1 \le R, C \le 50$，最多 100 组询问。
算法讨论	雷必须直接点击，所以先统计好。 回到正常方向的扫雷游戏，我们把其它方块分为有数字与无数字两种（也就是周围有雷或无雷），每个无数字方块组成的连通块我们称之为一大块，则一大块中的方块会同时打开。而每个有数字方块，我们要么点击它自己、要么点击它周围的大块中的无数字方块。 注意到有数字方块周围至多有两个大块，对于这种状况，在反向游戏中，我们可以点击该有数字方块以使这两个大块同时关闭。所以我们构造一个无向图，用点来代表大块，对可同时打开的大块连边。那么这个图中每一个匹配都相当于减少了一次点击次数。我们直接利用带花树算法求最大匹配就解决了（注意对于不和无数字方块相邻的有数字方块我们还需要额外点击）。
时空复杂度	单组询问：时间复杂度 $O(S^3)$；空间复杂度 $O(S)$。S 为大块个数。

No. 9

试题编号	Codechef JULY 11
试题名称	Billboards
题目大意	对于长度为 N 的二进制串，需要保证任意连续的 M 位中至少有 K 个 1，并且整个序列中 1 的个数最少。问有多少种可能的串（对 1000000007 取模）。 数据范围：$1 \le K \le M \le 50$，$M \le N \le 10^9$。
算法讨论	考虑 M 整除 N 的情况，我们把串均分成 N / M 段。于是每一段中至少有 K 个 1，整个串中至少有 K * (N / M)个 1。不难看出，下界是能够取到的。 记第 i 块中第 j 个 1 在块中的位置是 $block_{i,j}$，于是对于任意 $x < y$，满足 $block_{x,z} \ge block_{y,z}$。将 block 按列优先写成矩阵的形式，恰为半标准杨氏矩阵。其计算公式为：$\dim = \prod_{(i,j)} \frac{r+j-i}{hook(i,j)}$，其中 hook(i, j)表示矩阵中同行或同列，纵坐标或横坐标大于(i, j)的个数。r 为候选数集合大小。 由于答案的分子分母可以约分，有效个数是 O(KM)。 对于非特殊情况：若 $N \% M \le M - K$，那么每组前 N % M 个位置都应该是 0；若 $N \% M \ge M - K$，那么每组后 M - N % M 个位置都应该是 1。转化后与特殊情况相同。
时空复杂度	时间复杂度 O(KMlogMod)；空间复杂度 O(1)。

No. 10

试题编号	Codechef JULY 11
试题名称	Trial of Doom
题目大意	在一个 n * m 的方格内，有 k 个方格是红色、其余为蓝色。当前在(1, 1)，出口在(n, m)，每次可以移动到八个相邻方格中，每次离开一个方格，这个方格及

	其周围四个方格会改变颜色。给定房间颜色情况，问能否将所有方格都变成蓝色的情况下离开房间。 数据范围：$1 \le t \le 50$，$1 \le n, m \le 10^9$，$\min(n, m) < 40$，$0 \le k \le \min(m * n, 10000)$。
算法讨论	①若 $\min(n, m) > 1$，易证我们可以在不影响其它方块的情况下，对一个方块及其四周的方块取反，并在任意一个方块停下。于是原问题转化为了一个经典的可以通过解异或方程组解决的问题。注意到 $n * m$ 可能很大，但 $\min(n, m)$ 很小，不妨设行数小于列数。对于每一个红色的方块，我们可以通过在前一列操作逐步的将其移动到第一列中。那么最后只要解第一列能否通过操作最后一列将其消为全蓝即可。若某一列的某一行有红色则前一列、前两列、前三列……需要按哪些位可以预处理出来，通过找循环节能够加速这一过程。 ②若 $\min(n, m) = 1$，我们仍设行数小于列数。对于每个红色块，我们可以发现，假如它到终点的距离在模 3 意义下为 2，则其对答案是无贡献的；否则记 dist % 3 为 0 的红色块数目为 a、dist % 3 为 1 的红色块数目为 b，则答案只与 a % 2、b % 2、m % 2、m % 3 有关。
时空复杂度	单组时间复杂度 $O(K)$或 $O(K + \min(n, m) * L + \min(n, m)^3)$；空间复杂度 $O(K + \min(n, m) * L)$。L 为循环节长度

No. 11

试题编号	Codechef AUG 11
试题名称	Shortest Circuit Evaluation
题目大意	布尔型表达式（变量和与、或、非）满足短路运算原理，若值已确定则不再计算多余项。 现在给出表达式，和每个变量（互相独立）为真的概率，试调整布尔型表达式的顺序，使得期望的计算次数（使用变量的次数）最少，求期望概率。 本题数据规定将表达式化简后与原表达式相比无实质影响。 数据范围：表达式长度不超过 30000 个字母，表达式不超过 1000 个变量。
算法讨论	我们可以先把表达式看成树形结构（不一定真的建出一棵树来）。由于题目保证化简后无影响，所以其中的调换只能是一组 and 相连的对象或一组 or 相连的对象。那么对每一个树上的结点，我们可以记其正确的概率 P 及期望计算的次数 E。 以父结点 and 所连接的 t 个孩子为例（记为 $C_1, C_2, \ldots C_t$）： $P = \prod_{i=1}^{t} P_{C_i}$； $E = \sum_{i=1}^{t}(E_{C_i} \cdot \prod_{j=1}^{i-1} P_{C_j})$。 当 $x + 1 = y$ 时，推导出交换 x、y 后的期望 $E' < E$ 当且仅当 $E[C[y]] / (1 - P[C[y]]) < E[C[x]] / (1 - P[C[x]])$。故我们直接贪心即可。 对于父结点 or 所连接的对象，情况是类似的。
时空复杂度	时间复杂度 $O(S\log S)$；空间复杂度 $O(S)$。S 为表达式长度。

No. 12

试题编号	Codechef SEPT 11
试题名称	Short
题目大意	给定两个数 n、k，求满足 $n < a < k$、$n < b < k$，并且$(a - n)(b - n) \mid (ab - n)$的$(a, b)$对数。

	数据范围：$0 \le n \le 100000$，$k \le 10^{18}$。
算法讨论	令 $c = a - n$、$d = b - n$，原题改为求满足 $1 \le c, d < k - n$ 且 $cd \mid (c + n)(d + n) - n$ 的(c, d)组数。我们有 $d = \dfrac{cn + n^2 - n}{tc - c - n}$，其中 t 为整数。 设 $c < d$，推得 $c < 2.12n$。因而我们枚举 c，再枚举右式分子的约数并判断；当 c 较大时（例如大于 3000），因子较多而算出来的 t 较小，我们再改成枚举 t 来提升效率。
时空复杂度	时间复杂度 O(玄学 * n)；空间复杂度 O(nlogn)。

No. 13

试题编号	Codechef SEPT 11
试题名称	Counting Hexagons
题目大意	求满足元素均为在[1, N]中的整数、最大数不小于 L、其它数不大于 X、同样大小数不超过 K，且能组成六边形的无序六元组数目。 数据范围：$2 \le N \le 10^9$，$2 \le L \le N$，$N - L \le 100$，$1 \le X < L$，$1 \le K \le 5$。
算法讨论	由于 N - L 很小，很显然可以枚举最大边长度 M，则问题转化为求和大于 M、最大数不超过 X、相同数个数小于 K 的单调不下降的五元组个数。 接下来我们可以使用数位 DP。状态需要记录： ①从低位向高位枚举，当前在第几位； ②升序排列后，每相邻一对数是否相等； ③就之前枚举过的这些位而言，五个数的和与 M 的关系； ④就之前枚举过的这些位而言，每个数与 X 的大小关系； ⑤五个数的和对当前枚举位的进位。 转移时即枚举五个数的当前位是 0 还是 1，我们可以预处理不包含当前位的信息，每一种状态的每一种转移后的状态。由于数不能为 0。我们可以事先对 X 减一、对 L 减五。
时空复杂度	时间复杂度 O(St * Tr * (N - L)logN)；空间复杂度 O(St * Tr + StlogN)。其中的 St 指状态数（约 480），Tr 指转移数（为 32）。

No. 14

试题编号	Codechef OCT 11
试题名称	The Baking Business
题目大意	给出 S 组操作，每组操作可能为出售或询问。 对于出售，形如： I 产品编号[.大小编号] 省编号[.城市编号[.地区编号]] 性别 年龄 出售数量 对于询问，形如： Q 产品编号[.大小编号] 省编号[.城市编号[.地区编号]] 性别 起始年龄[-结束年龄] 其中方括号为模糊选项。可能询问所有产品、所有省。 $S \le 100000$，I 操作不超过 100，年龄属于[1, 90]。
算法讨论	对于每组插入，直接用数组更新每一种可能出现的询问的答案，其中年龄信息可用前缀和存储。
时空复杂度	时间复杂度 O(S)；空间复杂度 O(St)。其中的 St 指可能出现的询问种数。

No. 15

试题编号	Codechef OCT 11
试题名称	Sine Partition Function
题目大意	求 $f(M,N,X)=\sum_{k_1+k_2+...+k_M=N}\sin(k_1X)\sin(k_2X)...\sin(k_MX)$ 。 数据范围：$M \le 30$，$N \le 10^9$，$0 \le X \le 6.28$。
算法讨论	利用三角恒等变换，我们得到，在各项均有意义时： f(m, n, X) = sinX * f(m - 1, n - 1, X) + 2cosX * f(m, n - 1, X) - f(m, n - 2, X)。 所以我们可以构造一个矩阵，存下相邻两个 n 对所有 m 的取值，用矩阵快速幂转移即可。
时空复杂度	时间复杂度 $O(M^3\log N)$；空间复杂度 $O(M^2)$。

No. 16

试题编号	Codechef NOV 11
试题名称	Lucky Days
题目大意	定义一个定义在模 P（P 为质数）同余系下的序列 S，满足： ①$S_1 = A$； ②$S_2 = B$； ③$S_i = X * S_{i-1} + Y * S_{i-2} + Z$。 给定 C 和 Q 个询问，回答在序列[L, R]的区间内，有多少个 C。 数据范围：$P \le 10007$，$Q \le 20000$，$1 \le L \le R \le 10^{18}$。
算法讨论	当 X、Y 均为 0 时，O(1)判断即可。否则当 Y 为 0 时，转移矩阵可以简化为二阶；Y 不为 0 时，转移矩阵为三阶。对于后两种情况，易证矩阵都是可逆的。我们先利用 BSGS（三阶时）或暴力（二阶时）求出序列的周期 T，并记下在序列一个周期内的哪些位置，其值为 C。那么对于形如[1, X]的询问，我们容易将 X 分成 a * T + b (b < T)，前半部分则直接统计、后半部分则二分。 对于[L, R]的询问，显然可以写成[1, R]与[1, L - 1]两个询问的差值。
时空复杂度	时间复杂度 $O(P^{1.5} + Q\log P)$；空间复杂度 $O(P^{1.5} + T)$。

No. 17

试题编号	Codechef DEC 11
试题名称	Short II
题目大意	给定 p（一个质数，小于 10^{12}），问有多少对 a, b(a > p, b > p)满足(a - p)(b - p) | ab。
算法讨论	令 c = a - p、d = b - p，题目的条件等价于 cd | p(c + d + p)。 若 p 整除 c 且 p 整除 d，满足此条件的 c / p、d / p 只有(1, 1)、(1, 2)、(2, 1)、(2, 3)、(3, 2)五对。 若 p 不整除 c 且 p 不整除 d，此时要么 c = d = 1，要么 $c \ne d$。我们设 c < d，有 $c<1+\sqrt{p+1}$ 。由于 d 可以写成 $\frac{c+p}{ck-1}$（k 为满足条件的整数）的形式，我们不妨设 t 为上式的分母，于是一组合法解需满足下列条件：c < d；c 不被 p 整除；t 整除 c + p；c 整除 t + 1。我们枚举 d 与 t 中的较小值，其上界均为

	$\sqrt{p+1+\sqrt{p+1}}$，如此即可找出所有的解。 若 p 只整除 c、d 中的一个，其对数恰为上一种情况对数的两倍。
时空复杂度	时间复杂度 $O(p^{0.5})$；空间复杂度 O(1)。

No. 18

试题编号	Codechef DEC 11
试题名称	Hypertrees
题目大意	定义三超图，相比于普通图，其每条边都连接三个点；同理可定义三超树，指去掉任意一条边以后都不连通的三超图。 给定 N($3 \le N \le 17$)，问有几种含有 N 结点的带标号三超树。
算法讨论	由于数据范围很小，我们选择打表。 对于一个点双连通的三超树中的一条边连接的三个点，有且仅有一个点只属于这条边（仅有三个点的情况除外）。删除这样的点后，将得到一个点双连通普通图，该图的点数加边数对应着原图的点数。 我们暴力求出这样的普通图的个数，则容易得到点双连通三超树的个数。继而暴力将这些点双连通分量组合成大的三超树就得到了原问题的答案。
时空复杂度	时间复杂度 O(1)；空间复杂度 O(N)。

No. 19

试题编号	Codechef JAN 12
试题名称	Card Shuffle
题目大意	有一个序列，其初始状态为首项 1、末项 N 且公差为 1 的等差序列。做 M 次操作，每次操作为： ①拿走前 A 个数； ②拿走当前的前 B 个数； ③将第一步拿走的 A 个数按原次序插入序列最前面； ④拿走当前的前 C 个数； ⑤将第二步拿走的 B 个数逆序插入序列最前面； ⑥将第四步拿走的 C 个数按原次序插入序列最前面。 数据范围：$1 \le N, M \le 100000$，保证操作合法。
算法讨论	使用带区间翻转标记的 Splay 来模拟所有操作。
时空复杂度	时间复杂度 O(MlogN)；空间复杂度 O(N)。

No. 20

试题编号	Codechef JAN 12
试题名称	Misinterpretation 2
题目大意	求长度在[L, R]中，满足偶数位字符依次取出与奇数位字符依次取出相接与原串相等的、由小写字母构成的字符串个数，对 1000000007 取模。 数据范围：$1 \le L \le R \le 10^{10}$，$R - L \le 50000$。
算法讨论	设原字符串为 S、字符串长度为 X，我们把变换后的字符串与原字符串对应位相等的关系写下来。当 X 为偶数时，$S_i = S_{2i} \bmod (X + 1)$；当 X 为奇数时，$S_i = S_{2i} \bmod X$ 对 $i < X$ 成立。 由于一一对应，我们可以把这种相等关系写成置换的形式，并设长度为 X 时

	置换中的循环数为 G(X)，则原问题答案显然为：$\sum_{i=L}^{R} 26^{G(i)}$。 不难看出当 X 为奇数时 G(X) = G(X - 1) + 1，故我们只考虑 X 为偶数的情况。 我们令 ord(d)表示 2 模 d 的阶，那么有 $G(X)=\sum_{p整除X+1并且p\neq 1}\frac{\varphi(p)}{ord(p)}$。通过枚举质因数法，我们可以在 O($R^{0.5}$ + (R - L)logR)的时间范围内批量的对[L, R]内的数质因数分解。由于欧拉函数是积性函数，ord 函数也满足 x、y 互质时，ord(xy) = lcm(ord(x), ord(y))，因而我们只需要求出质数及其幂的 phi 和 ord。若当前考虑质数 p，p 及 p 的幂的欧拉函数的求解是显然的，而 p 的 ord 可以从 p - 1 开始不断试除 p - 1 的质因子（这里的质因子可以直接用素数表求出）求得，求其幂的 ord 的思路也是类似的。对于一个 p 而言，以上过程的复杂度是 O($\log^2 p$)。 所有范围内的偶数 X，对 X + 1 批量分解质因数（分解过程中可能出现大质数，需要计算 phi 及 ord），再对 X + 1 做一次 DFS 求因子，直接按上述两个公式计算就能算出最终的结果。
时空复杂度	预处理时间复杂度 O($\frac{R^{3/4}}{\log^2 R}+\sqrt{R}\log^2 R$)； 单组时间复杂度 O($(R-L)(\frac{\sqrt{R}}{\log R}+\log^2 R)$)；空间复杂度 O($R^{0.5}$)。

No. 21

试题编号	Codechef FEB 12
试题名称	Find a Subsequence
题目大意	给定一个长度为 N 的数组 A，并给出一个 1~5 的排列 B，试在 A 中找出一长度为 5 的子序列，使这五个元素相对大小顺序与 B 相同。 数据范围：$N \leq 1000$。
算法讨论	枚举该子序列中第二个数和第四个数的下标位置，那么剩下三个数各自的位置范围就确定了。又因为我们已经知道这五个数的大小关系，所以也可以得到这些数值的大体范围。对于剩下的三个数，根据贪心的思想，最小的那个数应该尽量小、最大的那个数应该尽量大，如此则中间的那个数能选择的余地才更大。 因而我们的任务就在于在指定区间内、指定值域内，求最小或最大的数、询问是否有数。这可以通过预处理前缀位置、权值二维数组，然后二分来求解。
时空复杂度	时间复杂度 O($N^2\log N$)；空间复杂度 O(N^2)。

No. 22

试题编号	Codechef FEB 12
试题名称	Flight Distance
题目大意	有 N 个点、M 条边（无重边自环），每条边有一正权，可选择将其增加或减少一个值 d（保证这么做后依然为正数）并付出 d 的代价。现在可以对整幅图

	的权值进行修改，保证修改后任一条边两点间的直接距离不比经过中转点的距离长，试最小化代价。 数据范围：$N \le 10$，初始权值 ≤ 20。
算法讨论	容易把所有限制条件及最小化的目标函数均写成线性规划的形式，并使用单纯形来求解。具体做法就是：把每条边增加多少、减少多少，以及任意两点间最短路 $g_{i,j}$ 都设出来，然后根据题目要求“直译”。 但是这样做，在 CC 上会 TLE，原因是找可行解的过程过慢。我们可以尝试手动找一组可行解。显然，把所有边长都赋为相同值是合法的。由于初始权值不超过 20，易证存在最优解使得最终权值也不超过 20。我们设 $h_{i,j}$ 为两点间最少经过边数，有 $g_{i,j} \le 20h_{i,j}$，令 $g_{i,j} = 20h_{i,j} - g'_{i,j}$，并把 $g'_{i,j}$ 作为变量，则全 0 可行解满足所有边权均为 20，同理将增加、减少值用 19 - 新变量来代替，并增加原变量非负用新变量表达的不等式。即可很快通过所有数据。 最终答案要求输出分数，我们可以直接做浮点数运算，最后枚举分母来解决。
时空复杂度	时间复杂度 O(玄学)；空间复杂度 $O(N^4)$。

No. 23

试题编号	Codechef MARCH 12
试题名称	Evil Book
题目大意	有 N 个点，每个点有 C_i 点代价和 M_i 点权值。每付出 C_i 点代价，可以得到 M_i 点权值，并且删除该点。若当前拥有权值超过 X，可以选择消耗 X 点权值并把某一个点的 C 和 M 均乘上三分之一。初始时拥有权值为 0，问达到 666 点权值最少需要付出多少代价。 数据范围：$N \le 10$，$10 \le X < 666$，$0 \le C_i, M_i \le 10^9$。
算法讨论	设对第 i 个点，我们使用了 k 次魔法，则应满足 $\frac{M_i}{3^k} > kX$。我们设 d_i 为最少执行了 d_i 次魔法后，M_i 才小于 666。那么对这个点，在容许的情况下，我们最少应尽量执行 $d_i - 1$ 次魔法（否则多贡献的权值一定是多余的）。 那么，我们可以把当前每个点的状态（没有被删或用了 k 次魔法被删）记录下来，在可行性剪枝、最优化剪枝、记忆化的前提下进行搜索。通过取极限情况计算发现，每个点的状态不超过 r ($r = 4$)种。
时空复杂度	时间复杂度 $O(rNr^N)$；空间复杂度 $O(r^N + N)$。

No. 24

试题编号	Codechef MARCH 12
试题名称	Ciel and Earthquake
题目大意	有一方格网络，其中共 $R \times C$ 个交叉点。每两个四向相邻点间原先有边，每条边均有概率 p 断开。行、列下标均从 1 开始，求(1, 1)与(R, C)连通的概率。 数据范围：$R \le 8$，$C \le 10^{18}$，$0.1 \le p \le 1$。
算法讨论	模仿插头 DP 的思路，我们列优先枚举进行 DP。对于第 i 行第 j 列，其状态值应设定为：只考虑前 j - 1 列和第 j 列前 i 行的点及其之间的边，那么(1, j)一直到(i, j)、(i + 1, j - 1)一直到(R, j - 1)的这些轮廓线上的点互相的连通情况（记录每个点属于哪个连通块即可）。打表可知这样的合法状态数很少，对于 $R = 8$ 也只有 3432 种。转移即枚举新加入的点和老的点之间边的通断状况，这可以预处理出来。

	此外，此题的 C 范围很大，但是当 j 足够大（例如达到 50 后），我们发现在 R 不变的情况下，C = j 和 C = j - 1 的答案之间的比值趋向于一个常数。因而我们最多只做到 50，后面的快速幂即可。
时空复杂度	时间复杂度 O(min(50, C)RS)；空间复杂度 O(min(50, C)RS)。S 为状态数。

No. 25

试题编号	Codechef APRIL 12
试题名称	Substrings on a Tree
题目大意	给出一个 N 个结点的有根树，每个结点有一个小写字母。所有能通过深度从小到大的简单路径构成的字符串（包括空串）被称为这棵树的子串。首先回答这棵树有多少个子串，此外再回答 Q 个询问，每个询问包含一个 26 个字母的排列（给定这 26 个字母从小到大的顺序）和一个数 K，输出按这个顺序升序排列后，不同的子串中第 K 大的子串，若不存在则输出-1。 数据范围：$N \le 250000$，$Q \le 50000$，保证输出不超过 800KB。
算法讨论	直接对树 DFS，按照父子关系建立广义后缀自动机。在后缀自动机中每一个从 root 开始的合法路径均对应了整棵树的一个子串，且按照自动机的拓扑倒序遍历，可求出在自动机上能走到某一结点的字符串的个数 cnt。则首先回答的值就是自动机 root 的 cnt。 对于后面的询问，我们在自动机上走一遍即可。具体而言，对于当前结点、当前 K，我们按照输入的字符顺序遍历其孩子，按照孩子的 cnt 不断减小 K，或判定答案就在某个孩子当中时转移，边走边输出，当答案减到 1 时结束。
时空复杂度	时间复杂度 O(N + L)；空间复杂度 O(N)。L 为 Q 个询问答案的总长度。

No. 26

试题编号	Codechef APRIL 12
试题名称	Find a special connected block
题目大意	给出一个 N * M 的矩阵，每个位置填一个在[-1, N * M]中的整数，且均有一个选择代价（不超过 100000 的正整数）。试找出一个至少包含 K 个不同正数、且不含-1 的四连通块，要求最小化代价。 数据范围：$1 \le N, M \le 15$，$1 \le K \le 7$。
算法讨论	把矩阵中每种正数看成一个颜色，建立一个所有存在的颜色向[1, K]的满射。则此时做原问题，其代价一定不小于原问题的代价（并且最优解可以取到）。随机完颜色后，题目就成为了一个经典的斯坦纳树问题。我们可以设状态 F[i][j][k]表示当前在矩阵的(i, j)点，当前连通块中存在哪些颜色的二进制表示为 k，此时所要付出的最小代价。转移时考虑向相邻格子扩展和把一个格子为“头”的多个连通块合并。答案就是所有非障碍格子 F[i][j][2^K - 1]的最小值。
时空复杂度	时间复杂度 O(T2^KNM)，T 由调参得到；空间复杂度 O(2^KNM)。

No. 27

试题编号	Codechef MAY 12
试题名称	Little Elephant and Boxes
题目大意	有 n 个盒子，打开第 i 个盒子，有 P_i / 100 的概率获得 V_i 元钱，否则获得一个钻石。又有 m 个物品，第 j 个物品（只有一个）需要花费 C_j 元钱和 D_j 个钻石。假设总是最大化买的物品数量，试计算打开所有盒子后，期望购买的物品个

	数。 数据范围：$2 \le n \le 30$，$1 \le m \le 30$，$0 \le D_j \le 30$，$1 \le V_i, C_j \le 17$。
算法讨论	我们设 cost[i][j]表示购买 i 件物品，使用不超过 j 个钻石，最少需要的钱数。这很显然可以通过枚举物品，通过以下方程转移预处理： cost[i][j] = min(cost[i][j], cost[i - 1][j - D[k]] + C[k])。 考虑 meet-in-the-middle，我们令 a + b = n。先考虑大小为 b 的后半部分，枚举每次获得钱或是钻石，由于钻石数较少，我们把每一种可能的钱数、概率二元组都存在对应的以钻石个数为第一关键字的一组 vector 中。接着对每一个 vector 中的元素均以钱数为关键字排序。 再考虑前面大小为 a 的部分，依旧枚举所有可能获得钱、钻石的情况，并枚举我们在后半部分中获得了多少钻石，那么假如我们要恰好买 cnt 个物品，则我们获得的总钱数必须在[cost[cnt][钻石数], cost[cnt + 1][钻石数])内。显然这在我们枚举的钻石数对应的 vector 序列中是一段连续区间，可以二分得到。直接统计其概率和算贡献即可。
时空复杂度	时间复杂度 $O(nm + b2^b + mb^2 2^a)$，a 可取 0.4n；空间复杂度 $O(2^b)$。

No. 28

试题编号	Codechef MAY 12
试题名称	Selling Tickets
题目大意	有 n 道菜，m 个顾客。m 个人每个人都有两道不同的特别喜欢的菜。每道菜只能供应一位顾客，且每个来的人必须至少得到一道特别喜欢的菜。厨师想卖出尽可能多的票，且满足无论哪些人得到票都能满足要求。求最多卖出的票数。 数据范围：$2 \le n \le 200$，$0 \le m \le 500$。
算法讨论	将 n 道菜看成点，m 个人看成无向边，那么能卖出 K 张门票的充要条件是该图中任意大小为 K 的边集 E，其对应的点集 V 满足 $K \le \|V\|$。那么原问题就是找一个最小的边集 E，使得$\|E\| > \|V\|$，此时答案为$\|E\| - 1$，并且必然满足$\|E\| = \|V\| + 1$。通过简单的讨论得到，该图的结构只有这三种可能： ①两个度数为 3 的点之间有三条不相交的路径； ②两个不相交的环间由一条简单路径连接； ③两个环在一个顶点处相交。 第一种情况我们枚举起点，终点，找出其间最短的三条路径；对于二、三，我们找从一个点出发的最大环和次大环，分别通过路径或点组装到一起即可。以上枚举时不合法的情况一定能转化合法、或不是最优。
时空复杂度	时间复杂度 $O(n^3 + n^2m)$；空间复杂度 $O(n^2)$。

No. 29

试题编号	Codechef JUNE 12
试题名称	Cool Numbers
题目大意	一个十进制数 n，设其数位和为 Q，若该数存在 1 到 3 个数位上的数字和为 S，且$(Q - S)^S$是 n 的倍数，那么 n 是 cool number。T 组询问，每次给定 N，分别求不大于 N 最大的和大于 N 最小的 cool number。 数据范围：$T \le 100000$，$1 \le N \le 10^{1000}$。
算法讨论	若一个数只有 1 到 3 位不是 0，则其必然满足条件；否则设其位数 k，必然满

	足$(9k)^{27} \ge 10^{(k-1)}$，这样 k 范围足够小。 我们发现后者，其数位和也是有限的，我们枚举每一种数位和，再暴力搜索哪些数满足条件，打表可知不超过 40000 个。我们预处理并将其排序。每次询问时，我们可以先 O(位数)的简单推算出前一种数，再在后一类数中二分，综合求最优即可。
时空复杂度	时间复杂度 $O(\text{玄学预处理} + T(\log N + \log M))$；空间复杂度 O(ML)。M 为第二类数的个数，L 为第二类数最大长度。

No. 30

试题编号	Codechef JUNE 12
试题名称	Expected Maximum Matching
题目大意	有一二分图，左边点数 n，右边点数 m，且左边第 i 个点与右边第 j 个点间有边的概率为 f[i][j]，求其期望匹配。 数据范围：$n \le 5$，$m \le 100$。
算法讨论	根据 Hall 定理，左边集合完全与右边集合匹配的条件是：左边任意子集 A 至少与|A|个右边结点相邻。令 f[i][S]表示只考虑右边前 i 个点，左边所有可能的点集合是否满足条件的情形为 s。可以看到 s 的理论个数达到 2^{32}。但是事实上，合法的个数远远达不到这么多，设 S 为 s 的有效个数，由打表可知 S 在 4000 左右。我们枚举有效的状态（可利用 map 或 unordered_map），再枚举左边点与右边枚举当前点的连通情况来转移。
时空复杂度	时间复杂度 $O(m2^nS)$；空间复杂度 O(mS)。

No. 31

试题编号	Codechef AUG 12
试题名称	Two Magicians
题目大意	一个 n 个点、m 条边的简单无向图上有两个魔法师。初始时第一个魔法师在 1 号点、第二个在 2 号，每个都有 P 点法力。第一个魔法师先走，每个魔法师在自己回合要做三件事： ①从当前房间沿着已有的路随意走，最终停在某个房间。若他恰停在另一个魔法师的位置，则获胜，否则继续； ②增加一条当前没有的路，若做不到则失败； ③若他有法力值，他可以选择消耗一点并传送到任意位置。 假如两个魔法师均使用最优策略，问谁会获胜。 数据范围：$2 \le n \le 7777$，$m, P \le 10000$。
算法讨论	注意到当前游戏的状态只与两人所在连通块大小的奇偶性、当前剩下的传送次数、大小为奇数的块的个数、大小为偶数块的个数、添加后不影响连通性的边的个数有关。直接暴力转移就可以 DP，但复杂度过大。 进一步观察我们发现，每个人至多使用一次传送；固定其它量，奇数块数足够多（10 以上）时 DP 值 4 个一循环；固定其它量，偶数块数足够多（10 以上）时 DP 值不变。如此状态数就大大减小了。 DP 过程预处理好，每次询问只需要计算出各种类型的信息 O(1)查询。
时空复杂度	时间复杂度 $O(n\alpha(n))$；空间复杂度 $O(n + m)$。

No. 32

试题编号	Codechef AUG 12
试题名称	A Game of Thrones
题目大意	有两个人用 N 种数字（每种数最多 10^{18}，最多 10^9 个）轮流玩游戏。第一个人先选择一个数字作为当前数字，之后每个人在其回合，将当前数字从纸上擦去，然后选择一个和当前数字只差一个质因子的数作为当前数字。无法操作者失败。问是否先手必胜。若先手必胜还需输出他在第一轮选择的保证其获胜的最小的数。 数据范围：$1 \le N \le 500$。
算法讨论	将数字看成点，若两个数间能够互相转变，我们在其之间连边。这一步使用 Miller-Rabin 暴力。于是第二个人获胜当且仅当该图存在完备匹配。因为此图一定是二分图（可按照质因数分解的项数奇偶性讨论），增加源汇后跑网络流就能够解决第一问。 当先手必胜时，如果某初始数字能使其获胜，必定存在一个最大匹配使得这个数没有匹配边，我们可以枚举、更改流量并判断。每一步不需要完全重做，使用退流即可。
时空复杂度	时间复杂度 $O(N^3\log N)$；空间复杂度 $O(N\log N)$。

No. 33

试题编号	Codechef SEPT 12
试题名称	Knight Moving
题目大意	有一无限大方格棋盘，要从(0, 0)格移动至(X, Y)格。每一步只能从(u, v)格移动到(u + Ax, v + Ay)或(u + Bx, v + By)。此外，棋盘上有 K 个不能进入的障碍格。问有多少种移动的方案。 数据范围：$0 \le K \le 15$，坐标绝对值不超过 500。
算法讨论	若两个移动向量线性无关，则 A 和 B 各自被使用的次数是确定可解的。我们使用组合数计算并容斥掉障碍格就好了；若两向量线性相关，则这两个向量需要与(X, Y)同方向才有解，而且此时原问题转化为了一维问题。我们在一维问题上从两个方向轮流做 DP，不停的更新从原点到达每个点的移动方案数。假如我们做了区间大小（500）级别次的 DP 后目标点的移动方案数依然在增加，则我们判断方案无穷。
时空复杂度	时间复杂度 $O(K^2 2^K$ 或 $D^2)$；空间复杂度 O(K 或 D)。D 为坐标范围。

No. 34

试题编号	Codechef SEPT 12
试题名称	Annual Parade
题目大意	有一 N 个点、M 条边的带权有向图。无自环，但可能有重边。现在要用路径（至少含一条边）去覆盖这些点，路径数目任意，一个覆盖的权值如下定义： ①每当使用某条边一次都需要花费边的权值； ②若某条路径不是环则需要多花费 C； ③若某个点不被任意路径覆盖，则也要花费 C。 K 组询问，每组都给定一个 C，计算此时的最小代价。 数据范围：$2 \le N \le 250$，$1 \le M \le 30000$，$1 \le K \le 10000$。边权与 C 均为不超过 10000 的正整数。
算法讨论	类似最小路径覆盖，我们将原图拆点，从左边的点向右边的点连费用为其原

	图最短路权值（Floyd 求出）、流量为 1 的边。容易看出此时 Mincost + C * (N - Flow)就是我们要最小化的目标代价。改写上式，一次增广的贡献即为 Cost - C，对于确定的 C 我们不停增广，假如贡献为负数，则我们继续做费用流、否则停止。对于多组数据，我们发现每次费用流的贡献值是递增的，因而我们一开始做 N 次增广，将每次的 Cost 记下来。询问时二分求出在何时 Cost 不小于 C，维护前缀 Cost 代入公式计算即得答案。
时空复杂度	时间复杂度 $O(N^3 + K\log N)$；空间复杂度 $O(N^2)$。

No. 35

试题编号	Codechef OCT 12
试题名称	Max Circumference
题目大意	给出三角形 ABC 三个顶点的坐标，以及 N 个操作，第 i 个操作可以使 A 的 x 坐标增加 x_i、y 坐标增加 y_i。最多使用 K 个操作（可叠加、不可重复使用），最大化三角形 ABC 的周长，答案的绝对误差必须小于 10^{-12}。 数据范围：$K \le N \le 500$，顶点坐标绝对值在 10^9 内，操作坐标绝对值在 10^6 内。
算法讨论	|BC|固定，问题即为最大化|AB| + |AC|。可以证明，存在两个数 u、v，使得我们相当于在最大化 $uA_x + vA_y$。 若 u、v 已知，直接对每个操作的贡献排序，取在前 K 中且非负的所有操作就可以了。若把(u, v)看成向量，则任意两个操作的相对顺序在(u, v)旋转一周的过程中只会发生两次改变，任意一个操作的贡献取反也只会发生两次。我们把发生这样事件时的向量看成关键向量，对其极角排序，那么就可以每次 O(1)的维护操作贡献的有序序列以及我们要选用的操作。注意当多个事件同时发生时需要小心处理。 此外，这题的精度要求较高。我们不能直接使用 sqrt 函数。设 $S^{0.5} = I + F$（I、F 分别指代整数、小数部分），需要用 $F = \dfrac{S - I^2}{I + \mathrm{sqrt}(S)}$ 来计算。
时空复杂度	时间复杂度 $O(N^2\log N)$；空间复杂度 $O(N^2)$。

No. 36

试题编号	Codechef OCT 12
试题名称	Ciel and password cracking
题目大意	Ciel 有 K 个密码需要暴力破解，第 i 个密码是一个不超过 N_i 的正整数。市里共 C 个电脑中心，她打算使用其中 K 个。这些电脑中心和 Ciel 的初始位置都在数轴上，若以 Ciel 的起始位置为零点，则第 i 个电脑中心的位置为 X_i。Ciel 的行走速度为每个单位时间 V 个单位坐标。 在第 j 个电脑中心，Ciel 可以使用最多 P_j 个电脑，每台电脑需要消耗 T_j 单位时间去验证一种密码。她先不消耗时间的选择一部分电脑，并在一台电脑中安装程序，在一段时间后这些电脑均安装了程序，此时同时开始破解，若有一台电脑成功破解密码，则她会立刻停止并奔赴下一个位置。 破解程序：每次随机验证一个没有被验证过的密码，消耗 T_j 单位的时间，不同电脑独立； 网络连接：每台电脑有一个连接列表，每次选择列表上第一个没有安装破解程序的电脑进行连接。连接所需时间随机，指数分布，均值为 S_j。程序传输、

	安装时间忽略不计。 数据范围：$1 \le C \le 1000$，$1 \le K \le \min(5, C)$，$1 \le V, S_j, T_j \le 10^{20}$，$1 \le N_i, P_j \le 10^{18}$，$\lvert X_i \rvert \le 10^{20}$。
算法讨论	令 Time[i][j]为第 i 个密码在第 j 个电脑中心破解的最小期望时间。我们设函数 F，并令 Time[i][j] = $F(N_i, P_j, S_j, T_j)$。则我们的任务为计算 F(N, P, S, T)。 设 A(k)为 S = 1 时，连接 k 台电脑的最小期望时间；B(N, k)为 T = 1 时，用 k 台点电脑破解范围为 N 的密码的期望时间，不难看出： $F(N,P,S,T)=\min_{k=1}^{P}(S\cdot A(k)+T\cdot \mathrm{B}(N,k))$。 先看 A(k)如何计算。由于指数分布的无记忆性，A(k) 为调和数列前 k - 1 项的和。那么对于 k 较小（如小于 10^6 时）我们可以暴力计算，否则用 $\ln(k-1)+\gamma$ 来近似。 对于 B(N, k)，由期望的线性性得： $B(N,k)=\sum_{i=1}^{N}\left(\frac{i}{N}\right)^{k}$。 当(k + 1) / N 较小时，我们对内层指数式套用欧拉-麦克劳林公式，并取阈值 p = min(10, k / 2)，有 $B(N,k)\approx\frac{N}{k+1}+\frac{1}{2}+\sum_{i=1}^{P}\frac{B_{2i}}{(2i)!}\prod_{j=0}^{2i-2}\frac{k-j}{N}$（右式的 B 指伯努利数）；否则大多数项较小，我们取较大的几个 i 按原式计算即可。 回到 F 函数的计算，当参数固定时，上面 F 的计算式中 min 内部以 k 为自变量的函数（设为 G）是单峰的，我们差分之后二分可以快速找到极值。 由于精度问题，我们还要再做一点讨论。指数函数$(1-a)^x$我们通常用 $\exp(x\ln(1-a))$来计算，但是当 a 非常接近 0 时会有很大误差，此时我们可以把 $\ln(1-a)$泰勒展开，取前两项来代替。此外对 G 函数差分时当中的 B 函数也不能直接差分，依旧模仿上述对 B(N, k)的计算，得到(k + 1) / N 较小时： $B(N,k)-B(N,k+1)\approx\frac{N}{(k+1)(k+2)}+\sum_{i=1}^{P}\frac{B_{2i}}{(2i)!}\left(\prod_{j=0}^{2i-2}\frac{k-j}{N}-\prod_{j=0}^{2i-2}\frac{k+1-j}{N}\right)$。 较大时也和上面类似处理。如此即可保证二分时需要的精度。
时空复杂度	时间复杂度 $O(CK\log P+2^KCK)$；空间复杂度 $O(CK+2^KC)$。

No. 37

试题编号	Codechef NOV 12
试题名称	Arithmetic Progressions
题目大意	给定一个长度为 N 的序列 A，求有多少个长度为 3 的子序列（可不连续）等差。 数据范围：$3 \le N \le 100000$，序列中的数为[1, 30000]内的整数。
算法讨论	{A[i], A[j], A[k]}为等差数列的条件是 2A[j] = A[i] + A[k]。我们对序列分块，按以下几种情况讨论： ①A[j]与 A[i]或 A[k]在同一块中，那么预处理前（后）缀的所有块中权值的计数数组，然后枚举同在当前块中的两个数统计； ②A[i]、A[j]、A[k]在同一块中，和上一种情况是类似的；

	②A[i]、A[j]、A[k]均不在同一块中，枚举 A[j]所在块，将前缀块和后缀块的权值计数数组做卷积，代入每一个 2A[j]查询。 块大小取 $O(N^{0.5}\log N)$时效率较高。
时空复杂度	时间复杂度 $O(N\sqrt{N\log N})$；空间复杂度 $O(N^{1.5})$。

No. 38

试题编号	Codechef NOV 12
试题名称	Martial Arts
题目大意	两支分别有 N 个选手的队伍两两配对比赛。已知任意两个人比赛的赛果（两队分别获得多少分）。记主队得分 H，客队得分 G，则主队想先最大化 H - G，其次最大化 H；客队则先最大化 G - H，其次最大化 G。赛程由主队安排，客队可以选择使得某一个选手所在的比赛不进行。假设客队一定会选择最优策略，问主队该如何安排比赛。 数据范围：$1 \le N \le 100$，单轮比赛得分为不超过 10^{12} 的非负整数。
算法讨论	将两支队伍看成二分图的两个点集，比赛看成边，我们定义边权为主客队获得分数的差值。则主队任务为最大化权值和，其次最大化 H；客队任务为最小化权值和，其次最大化 H。 不难看出，当匹配完成后，客队一定会删除当前匹配中权值最大且为正的那条边，则当我们将边升序排序后加入图中时，新加入的边会被优先删除。若我们强制匹配这条边，价值和就是除去这条边所连的点之外的最大权匹配。 最大权匹配问题我们往往用 KM 算法来解决。对于此题，一开始我们令每条边的权值为负无穷，在加入边时先令其权值为正无穷，算出此时的答案后，再将其变为应有的价值，重新维护最大匹配以等待下一条边加入。 由于我们每次只更改了一条边(u, v)，我们可以将左边点集中的 u 和其之前的匹配点在匹配集中删除，将 u 的顶标赋为边权减去 v 的顶标，只需要从 i 点出发找一条增广路就能高效地维护最大权匹配了。 由于我们还要同时最大化 H，因此直接将各种权值看成双关键字即可。注意在排序的时候当第一维相等时，第二维应当按降序。
时空复杂度	时间复杂度 $O(N^4)$；空间复杂度 $O(N^2)$。

No. 39

试题编号	Codechef DEC 12
试题名称	Different Trips
题目大意	一棵以 1 号点为根的 N 个结点的树，定义两个结点相似当且仅当其度数相同。定义可行路径为一个点到其某个祖先（含自己）的路径，且可行路径相似当且仅当其长度相同、按顺序一一对应的城市都相似。问有多少种不相似的可行路径。 数据范围：$1 \le N \le 100000$。
算法讨论	将度数看成字符，则任意结点返根的路径都是一个字符串，且可行路径为这些字符串的子串。问题等价于求 N 个字符串中不同的子串个数。与后缀数组求法类似，我们倍增求出每个结点向上 2^i 个结点对应的 N 个字符串的排名，直到能区分的都区分开。那么答案即所有字符串的长度和减去相邻两个排名的字符串的 LCP（LCP 亦可倍增求出）。

时空复杂度	时间复杂度 O(NlogN)；空间复杂度 O(NlogN)。

No. 40

试题编号	Codechef JAN 13
试题名称	A New Door
题目大意	给出一宽 W、高 H 的矩形。一开始全白色，向其中加入 N 个圆（位置不一定完全被矩形包含）并将每个圆的区域涂成黑色。最后将涂黑的区域挖去，求被挖去部分在矩形内的周长。 数据范围：$1 \le W, H, N \le 1000$。圆的坐标数值在[0, 1000]内，半径在(0, 1000]内，所有圆不完全相同。
算法讨论	若整个圆在矩形外或被另一个圆完全包含，则这个圆可以直接删除；否则我们记录每个圆被矩形框或其它圆切去的极角区间（对于跨过极角边界的情况需要增加两个区间），最后对每个圆处理，排序后容易算出其共被切去了多少极角。那么剩下部分的极角对应的周长就是该圆对答案的贡献。
时空复杂度	时间复杂度 $O(N^2 \log N)$；空间复杂度 $O(N^2)$。

No. 41

试题编号	Codechef JAN 13
试题名称	Cucumber Boy and Cucumber Girl
题目大意	给定 B 个 n * n 的矩阵 A_i，对于数对(a, b) (a < b)定义 n * n 的矩阵 C，使得： $C_{i,j} = \sum_{k=1}^{n} A_{a,i,k} \cdot A_{b,j,k}$ 。 一个 1 到 n 的排列 P 是好的当且仅当至少存在一个 i 使得 C_{i,P_i} 为奇数，数对(a, b)是好的当且仅当好的排列有奇数个。询问好数对个数。 数据范围：$n \le 60$，$B \le 8000$。
算法讨论	对于 n = 1 的情况，数对是好的当且仅当 a、b 所对应的矩阵中唯一的数均为奇数，我们特判即可。 对矩阵 C，我们定义矩阵 D 为 C 在每个元素上加 1、且模 2。由行列式的定义易知(a, b)是好的当且仅当 det(D)为奇数。若我们在每个矩阵 A 末尾加一个全 1 列，则 $D = A_a \times A_b^T \pmod 2$ 。 考虑 det(D)，由 Binet-Cauchy 定理，令 $A_{i,j}$ 为第 i 个矩阵删掉第 j 列得到的矩阵，有 $\det(D) = \sum_{i=1}^{n+1} (\det(A_{a,i}) + \det(A_{b,i}))$ ，故下求所有的 $\det(A_{a,i})$。 我们对每一个矩阵 A_i 进行消元，若不满秩，则 $\det(A_{i,j})$一定为 0；否则 Ai 能被消成 n * n 的单位矩阵加一列，设消元后矩阵为 E_i。若加上的是第 k 列，那么： ①$\det(A_{i,j}) = E_{j,k}$ (j < k)； ②$\det(A_{i,j}) = 1$ (j = k)； ③$\det(A_{i,j}) = 0$ (j > k)。 直接统计即得答案，消元及统计过程可使用 bitset 优化。

时空复杂度	时间复杂度 $O(Bn^2 + B^2)$；空间复杂度 $O(Bn^2)$。

No. 42

试题编号	Codechef FEB 13
试题名称	Room Corner
题目大意	给定一 N * M 的网格图描述一个房间（轮廓均为水平或垂直线段），保证房间内格子四连通，且若线段的端点在房间内，则整条线段在房间内。现在，每个拐点（离 90 度内角最近的那个格子）都有一个小朋友。两个在相邻的角落的小朋友可以交换位置，它们同时沿房间边缘向对方以一格每单位时间移动。不含公共点的交换可以同时发生。给出 T 组询问，每次询问一对小朋友想要通过不断交换最终相遇的最少时间。 数据范围：$N, M \le 2500$，$T \le 10000$。
算法讨论	将房间边缘对应的小朋友的环和每对相邻小朋友之间的距离预处理出来，因为题目细节较多，沿边缘扩展时需考虑方向等因素。 对两个想要相遇的小朋友，我们枚举在这两个小朋友相邻前，他们被交换到什么位置，例如 s 被交换到 i 的位置、t 被交换到 i + 1 的位置，则当前的答案即为 max(cost(s -> i), cost(t -> i + 1)) + dist(i, i + 1) / 2。由于是环，还需要分两个方向考虑。
时空复杂度	时间复杂度 $O(NM + TN)$；空间复杂度 $O(NM)$。

No. 43

试题编号	Codechef FEB 13
试题名称	Observing the Tree
题目大意	给定一个 N 个结点的树，结点带权，初始值均为 0。M 次操作。每次操作为路径加等差数列、询问路径上点的权值和或将当前状态回滚。 数据范围：$1 \le N, M \le 100000$。
算法讨论	对树进行树链剖分，用线段树来维护每一条重链。在线段树中每个结点记录该区间将加一个首项为 A、公差为 B 的序列，那么多个修改操作可以简单的合并起来。由于要实现回滚，所以我们对线段树可持久化。
时空复杂度	时间复杂度 $O(M\log^2 N)$；空间复杂度 $O(M\log^2 N)$。

No. 44

试题编号	Codechef MARCH 13
试题名称	Little Elephant and Colored Coins
题目大意	有 N 种类型的硬币，第 i 种硬币价值 V_i 元、颜色 C_i、数量无限。Q 组询问，每次询问用这些硬币组成 S 元最多有多少种颜色，若不可能则输出-1。 数据范围：$1 \le N \le 30$，$1 \le V_i, Q \le 200000$，$1 \le C_i \le 10^9$，$1 \le S \le 10^{18}$。
算法讨论	若只询问是否存在可行的方案，则我们可以选取最小的 V（设为 p），在模 p 的意义下 DP。令 f[i]为能拼成的模 p 为 i 的最小面值，那么直接判断 S 和 f[S % p]的大小关系即可。 此题在这一基础上还要最大化颜色数，因而我们对 DP 增加一维，用 f[i][j]表示使用 i 种颜色的硬币可以拼成的模 p 为 j 的最小面值就能够解决。我们每次考虑同一种颜色的硬币序列对 f 的贡献，注意在每次按第一维分层从大到小计算，转移时可以将整个长度 p 分成数个循环处理。

时空复杂度	时间复杂度 $O(N^2p)$；空间复杂度 $O(Np)$。

No. 45

试题编号	Codechef APRIL 13
试题名称	String Query
题目大意	给定一个初始长度为 10 的字符串 S，N 次操作，包括以下类型： ①在首、尾或正中间插入一个字符； ②在首、尾或正中间删除一个字符； ③给定一个字符串 T，问 T 在 S 中出现了多少次。 数据范围：$N \le 150000$，询问串长度和不超过 1500000。
算法讨论	将两棵反向的后缀平衡树接在一起形成加强版后缀平衡树，使之支持首尾的插入或删除操作。再把 S 串均分成两份，分别用加强版后缀平衡树来维护。每次操作后进行调整，使得 S 串始终被均分以支持中间的插入、删除。若一棵加强版平衡树的两棵子后缀平衡树为空，则重建该加强版平衡树，这样做的均摊复杂度是 $O(\|S\|\log\|S\|)$的（$\|S\|$取操作过程中最大值）。 对于询问操作，我们分情况讨论。对于在一棵后缀平衡树中的情况，我们直接在平衡树中查询小于它的最大的后缀和不大于它的最大的后缀之间有多少后缀；对于跨平衡树的情况，我们分别取出上一平衡树的后$\|T\| - 1$ 个字符和下一平衡树的$\|T\| + 1$ 个字符，暴力做 KMP。
时空复杂度	时间复杂度 $O(\|S\|\log\|S\| + \|T\|\log\|T\|)$；空间复杂度 $O(\|S\| + \|T\|)$。这里的$\|S\|$为最大值，$\|T\|$为总和。

No. 46

试题编号	Codechef MAY 13
试题名称	Queries on tree again!
题目大意	给定一个 N 个结点的带权（权在边上）基环树，保证环长为奇数。Q 组操作，包含最短路取相反数、最短路取最大连续子段和。 数据范围：$N, Q \le 100000$。
算法讨论	将边权转化为点权，用 DFS 找出环，删掉环上的任意一条边，记该边两端结点为 U、V。 此时，剩下的部分就是一棵树，我们对其树链剖分，用线段树来维护重链上的最大连续子段和。我们定义一条路径上的最大子段和相关信息为 ans(u, v)，重载加号来实现两段相连路径的合并，那么很容易通过树剖来求出任意一条树上路径的 ans(u, v)。考虑询问 s、t，若树上路径即为最短路则答案为 ans(s, t)；否则答案为 ans(s, U) + ans(V, t)（U 和 V 可能需要交换）。修改操作也是类似的。
时空复杂度	时间复杂度 $O(Q\log^2N)$；空间复杂度 $O(N)$。

No. 47

试题编号	Codechef JUNE 13
试题名称	Two k-Convex Polygons
题目大意	给定 N 个棍子的长度和整数 K。求能否在其中选出 2K 个棍子拼成两个各有 K 条边且相邻边不共线的凸多边形。 数据范围：$2K \le N \le 1000$，$3 \le K \le 10$。

算法讨论	K 根棍子能拼出合法凸多边形的条件是：边的长度和 ＞2 * 最长边长。那么不难看出将可选边集排序后，选出的边一定是连续的 K 个。 因为此题 K 很小，所以合法的方案有很多种。我们每次随机把 N 根棍子分到两个集合中，通过前缀和 O(N)检查两个集合是否存在合法方案即可。
时空复杂度	时间复杂度 O(TN)，T 由调参得到；空间复杂度 O(N)。

No. 48

试题编号	Codechef JUNE 13
试题名称	Count Special Matrices
题目大意	对于一个 N 阶方阵 A，若满足以下条件我们称其为特殊矩阵： ①$A[x][x] = 0\ (1 \le x \le N)$； ②$A[x][y] = A[y][x] > 0\ (1 \le x < y \le N)$； ③$A[x][y] \le \max(A[x][z], A[z][y])\ (1 \le x, y, z \le N)$； ④矩阵中的数均为[1, N - 2]内的整数，且这 N - 2 个整数均被取到。 T 组数据，每组给出 N，试统计 N 阶方阵中特殊矩阵的个数。 数据范围：$1 \le T \le 100000$，$3 \le N \le 10^7$。
算法讨论	打表（或推公式）得出答案等于 $\frac{N!(N-1)!}{3 \cdot 2^{N-1}}\left(\frac{3N}{2} - 2 - \sum_{i=1}^{N-1}\frac{1}{i}\right)$。 我们预处理出所有需要的量（必要时使用递推来降低复杂度），对于每个 N 套公式输出即可。
时空复杂度	时间复杂度 O(N + T)；空间复杂度 O(N)。

No. 49

试题编号	Codechef SEPT 13
试题名称	Two Roads
题目大意	给出平面上 N 个点坐标，作两条直线，每个点的权值为其到两条直线距离平方的较小值。最小化权值和。 数据范围：$N \le 100$，坐标范围均在[0, 1000]内，无三点共线。
算法讨论	两条直线夹角的两条正交的角平分线将平面分成四个象限，每条直线控制一组对角的象限。若我们知道所有将点分到四个象限的合法方案，那么我们可以直接根据Σx、Σy、Σxy、Σx^2、Σy^2 对每条直线控制的区域内的点直接推算出其最优直线的方程（该直线必过(|x|, |y|)点，设出方向参数求偏导）。 下面考虑划分点的方案。通过旋转与平移，我们一定可以把任意一对正交直线平移到以下状态：两个点分别从两个方向无穷接近一条线（即直线被这两点“夹住”）；另一条线被一个点无穷接近。因而我们枚举夹直线的两个点来确定一条直线，将其余点按照其在直线上的投影位置排序来枚举另一条直线的同时维护上述需要记录的量。
时空复杂度	时间复杂度 $O(N^3\log N)$；空间复杂度 O(N)。

No. 50

试题编号	Codechef SEPT 13
试题名称	To Queue or not to Queue
题目大意	给出一个初始为空的字符串 S。Q 次操作，或在 S 的末尾插入一个字符、或删

	除 S 的第一个字符。每次操作后，要求输出当前本质不同的子串个数（模 10^9+7）。 数据范围：$Q \le 1000000$，保证每次操作后字符串长度为正整数。
算法讨论	求子串个数，我们可以通过维护后缀树并求其每条边上的字符数量和来获得。若没有删除操作，则我们直接套用 Ukkonen 算法即可实现。 引入删除操作后，就需要我们对原算法略加拓展。注意我们只删除最长的后缀，我们记录每一个后缀的插入位置，在删除时删除这个插入位置所在结点，并合并所有度为 1 的祖先结点。注意要特判活跃结点所在位置。
时空复杂度	时间复杂度 O(Q)；空间复杂度 O(Q)。

No. 51

试题编号	Codechef OCT 13
试题名称	Fibonacci Number
题目大意	给定一个素数 P、非负整数 C，找到最小的非负整数 N，使得 $Fibonacci_N = C \pmod P$，若不存在输出-1。 数据范围：$11 \le P \le 2 * 10^9$，$0 \le C \le P - 1$。P % 10 为完全平方数，保证若存在答案，不超过 $2 * 10^9$。
算法讨论	根据题目条件，根号 5 在模 P 意义下存在。我们写出 Fibonacci 数列的通项公式，不妨设 N 为偶数，则问题转化为在模意义下求解方程：$x^N - x^{-N} = \sqrt{5}C$（x 为黄金数在模 P 意义下的倒数）。 不难解出，$x^N = \dfrac{\sqrt{5}C \pm \sqrt{5C^2+4}}{2}$。对于两个不同解，我们用 BSGS 尝试求解 N，判断是否存在及是否是偶数。对于 N 为奇数的情况也是类似的，最终输出两者答案的较小值。求一个数在模 P 意义下的平方根可以使用 Tonelli-Shanks 算法。
时空复杂度	时间复杂度 $O(\log P + P^{0.5})$；空间复杂度 $O(P^{0.5})$。

No. 52

试题编号	Codechef NOV 13
试题名称	Gangsters of Treeland
题目大意	给出一个 N 个点的树，0 号点为根。初始时每个点均被涂有不同的颜色。一条路径的代价被定义为过程中经过的点和上一个点发生颜色改变的次数。定义 f(u)为以 u 为根的子树中所有结点到根的路径的代价的平均值。Q 次操作，或询问一个点的 f 值，或将一个点到根的路径上所有点涂上一种相同的新颜色。 数据范围：$1 \le N, Q \le 100000$。
算法讨论	可以发现，修改操作与 LCT 中的 Access 操作十分相似，我们每次相当于给原先的偏爱孩子的子树结点的返祖代价加 1，更新的偏爱孩子的子树结点的返祖代价减 1。 我们扩展树状数组，使之支持区间加 1、减 1 操作，那么配合 DFS 序，我们就可以快速的询问子树返根的代价和。
时空复杂度	时间复杂度 $O(Q\log^2 N)$；空间复杂度 O(N)。

No. 53

试题编号	Codechef NOV 13
试题名称	Queries With Points
题目大意	平面上有 N 个互相没有公共点、且两两不包含的简单多边形。第 i 个多边形有 k_i 个点。Q 次询问，强制在线，每次给出一个点的坐标，判断其在哪个多边形内，或是不在任何多边形中。 数据范围：$1 \le N, Q \le 100000$，$3 \le \Sigma k \le 300000$。
算法讨论	从每个多边形的端点作 x 轴的垂线，将平面切成一个个长条。则每个长条中的图形都是一个或多个梯形（或退化的三角形）。对于每个询问点（特判在边界的情况）。找出其所在的长条后，用平衡树查询其上方第一条线段的下方属于哪个区域。 这样极端情况下我们需要维护 O(点数)棵平衡树。但是，对于每条线段，我们只有两个端点。若将维护线段的平衡树可持久化，我们发现在从左往右更新每个长条时，一条线段只有可能被插入和删除一次。因而使用可持久化 Treap，我们就能够事先预处理出每个长条的平衡树。
时空复杂度	时间复杂度 $O((\Sigma k + Q)\log\Sigma k)$；空间复杂度 $O(\Sigma k\log\Sigma k)$。

No. 54

试题编号	Codechef DEC 13
试题名称	Query on a tree VI
题目大意	给出一棵 N 个结点的树，每个结点可能为黑色或白色，初始全黑。M 组操作，或是询问与 u 同色的极大连通块有多大，或是将 u 的颜色反转。 数据范围：$1 \le N, M \le 100000$。
算法讨论	维护两个树链剖分结构分别处理黑、白两种情况。对于每个结点维护假定它是黑（白）色时，只考虑它的子树时它所在的极大连通块大小。当修改时，我们需要在两个树剖结构中分别修改其到对应色深度最小祖先的路径上的值；询问时找到与其同色的深度最小的祖先，在对应的树剖结构中询问其单点值。找深度最小的祖先同样可以在树剖得到的一些链上二分得到。
时空复杂度	时间复杂度 $O(M\log^2 N)$；空间复杂度 $O(N)$。

No. 55

试题编号	Codechef DEC 13
试题名称	Petya and Sequence
题目大意	给出一长度为 N、下标从 0 开始的序列 A，问是否存在一长度为 N、下标从 0 开始的序列 B，满足： ①B 中元素不全为 0； ②对于任意 j $(0 \le j \le N - 1)$满足 $\sum_{i=0}^{N-1} A_i \cdot B_{(i+j)\bmod N} = 0$。 数据范围：$1 \le N \le 30000$，$\lvert A[i]\rvert \le 1000$。
算法讨论	构造矩阵 C，使 $C_{i,j} = A_{(i+j)\bmod N}$，问题等价于判断 C 是否满秩。 定义 $f(x) = \sum_{i=0}^{N-1} A_i x^i$，那么若 $\gcd(f(x), x^N - 1)$的次数为 d，这个矩阵的秩即为

	N - d。再定义 $\phi_N(x)=\prod_{0\le i<N,(i,N)=1}(x-\omega_N^i)$，满足 $\prod_{d\|N}\phi_d(x)=x^N-1$。那么这样定义的多项式无法表示为两个次数均不为 0 的多项式的乘积，故只要判断是否存在 d|N 满足 $\phi_d(x)\mid f(x)$，这等价于 $f(x)\times\prod_{i\|d且i为质数}(x^{\frac{d}{i}}-1)$ 被 x^d - 1 整除。模拟判断即可。
时空复杂度	时间复杂度 O(NlogN + NS)，S 为 N 的因子个数；空间复杂度 O(N)。

No. 56

试题编号	Codechef JAN 14
试题名称	Counting D-sets
题目大意	一个点集的直径被定义为点集中最远的一对点的切比雪夫距离。若两组点集之间可以相互平移得到，则认为两点集相等。T 组数据，每组给定 N、D，求在 N 维空间的整点集合中，满足其直径恰好为 D 的等价类数，对 10^9+7 取模。数据范围：$1\le N\le 1000$，$1\le D\le 10^9$。
算法讨论	我们将问题放宽为小于等于 D 的点集个数，则答案为 $Ans_D - Ans_{D-1}$。为避免两个点集相同，我们可以限定每一维都存在坐标为零的点。那么所有点都被限定在原点附近一个边长为 D 的 N 维立方体中。预处理组合数，利用容斥和快速幂就可以计算了。
时空复杂度	时间复杂度 $O(N^2 + TN\log N)$；空间复杂度 $O(N^2)$。

No. 57

试题编号	Codechef JAN 14
试题名称	Counting The Important Pairs
题目大意	给定一个 N 个点、M 条边的简单无向连通图。要求删除两条边之后仍保持连通性，求删边的方案数（两条边无序）。 数据范围：$1\le N\le 100000$，$1\le M\le 300000$。
算法讨论	求出该图的一棵 DFS 树，对每条非树边随机一个权值，每一条树边的权值设为覆盖它的所有非树边的权值的异或和。那么对于一个边集，若删除它使原图不连通，一定存在一个子集使其边权的异或和为 0。当边权取在 long long 范围内时，这样做的错误率可以忽略不计。 由于要删除两条边，所以要么是两条边权值相等，要么是两条边中有一条权值为 0。对边权排序后，分两种情况计算即可。
时空复杂度	时间复杂度 O(N + MlogM)；空间复杂度 O(N + M)。

No. 58

试题编号	Codechef FEB 14
试题名称	Graph Challenge
题目大意	给定一个 N 个点、M 条边的有向图（无自环、重边），每个点编号为其 DFS 序。保证所有点均能从 1 号点到达。一个结点 x 对 y 来说是好的当且仅当 x < y 且存在一条 x 到 y 的路径使得中间结点编号都大于 y。一个结点 x 对 y 来说是最好的当且仅当它是所有对 y 的好结点中编号最小的。 Q 组询问，每次给出一个结点，问它是多少个结点的最好结点。

	数据范围：$1 \le N, Q \le 100000$，$M \le 200000$。
算法讨论	不难看出，一个结点的最好结点就是 Dominator Tree 算法中的半必经点。故直接做 Dominator Tree 算法即可。
时空复杂度	时间复杂度 $O(N\alpha(N))$；空间复杂度 $O(N)$。

No. 59

试题编号	Codechef FEB 14
试题名称	Count on a Treap
题目大意	维护一个大根堆 Treap，初始为空，N 次操作，要求支持： ①插入一给定关键字、随机权值的点； ②删除一给定关键字的点； ③询问两给定关键字结点间的距离。 数据范围：$1 \le N \le 200000$，保证任意时刻树中结点的关键字、随机权值两两不等。
算法讨论	Treap 的中序遍历就是权值顺序，容易发现两个结点的 LCA 就是它们对应区间中随机权值最大的结点，那么可以 $O(\log N)$地求出两点的 LCA。下面只需要求结点的深度。 观察发现一个序列中第 y 个结点是第 x 个结点的祖先当且仅当中序遍历两者区间内不存在权值比 y 大的结点。离散化关键字后使用线段树来维护一个结点左、右的祖先个数。例如在左边时，令函数 getl(a, b)为线段树第 a 个结点、当前最大值是 b 时这个区间范围内的祖先个数，通过维护区间最大值容易确定递归进入哪个子树询问，这一过程是 $\log(N)$的。 插入和删除都容易转化为权值修改，对于修改与询问，我们都需要到包含该单点的 $\log(N)$个结点中处理。
时空复杂度	时间复杂度 $O(N\log^2 N)$；空间复杂度 $O(N)$。

No. 60

试题编号	Codechef MARCH 14
试题名称	Chef and Graph Queries
题目大意	给定一个 N 个结点、M 条边的简单无向图。共 Q 组询问，每组给定 L、R，每次询问只考虑编号在 L 到 R 之间的边时，图中的连通块数。 数据范围：$1 \le N, M, Q \le 200000$。
算法讨论	按照 R 关键字对询问排序。使用 LCT 来维护由编号 1~R 的边形成的最大生成树（可能为生成森林）。其中，每条边的权值设为其编号。这样当我们发现新加入的边在同一个连通块内时，就可以将它们俩路径上编号最小的边删除，并用新边来替换。 对于询问[L, R]，我们查询当前 LCT 上权值大于等于 L 的边的数目，并设为 C，这可以用线段树或树状数组来维护。我们所求的答案就是 N - C。
时空复杂度	时间复杂度 $O((Q + M)\log N)$；空间复杂度 $O(N)$。

No. 61

试题编号	Codechef MARCH 14
试题名称	The Street
题目大意	维护两个长度为 N 的数组 A、B，Q 个操作，包括以下三种：

	①给定区间与 a、b，对区间内每个 i，令 A[i] = max{A[i], (i - l) * a + b}； ②给定区间与 a、b，对区间内每个 i，令 B[i] += (i - l) * a + b； ③给定 i，输出 A[i] + B[i]。 数据范围：$1 \le N \le 10^9$，$1 \le Q \le 300000$。
算法讨论	对于②操作，我们可以改写成 i * a' + b'的形式，其合并是显然的，直接用线段树来维护； 对于③操作，线段树中每个代表区间[l, r]的结点，我们记录一对数 a 和 b，代表对于任意 i ($1 \le i \le r$)，A[i]应当比 i * a + b 小。当新的一对 a'、b'操作于该结点时，由于这两个以 i 为变量的函数均为线性，因而最多具有一个交点。无交点时直接更改当前结点值或忽略本次操作后跳出；有交点只需要递归到它的一个孩子中操作。因为一个区间对应 O(logN)个结点，故一次更新的复杂度不超过 $O(\log^2 N)$。 对于询问，A 部分是显然的；B 部分我们从根[1, N]走到[i, i]，对路径上每组 a、b 形成的函数，代入自变量 i，取函数值的最大值。 本题 N 范围较大，可以使用动态开点来解决。
时空复杂度	时间复杂度 $O(Q\log^2 N)$；空间复杂度 O(QlogN)。

No. 62

试题编号	Codechef APRIL 14
试题名称	Chef and Tree Game
题目大意	有一棵 N 个结点的树，每条边有红或蓝的颜色。两个人轮流游戏，第一个人每次选择一条红色边删除，第二个人每次选择一条蓝色边删除，删除后和树根（1 号点）不连通的部分将被删除，最后不能操作的人算输。假设两人均使用最优策略，问两个人分别先手时谁获胜。 数据范围：$1 \le N \le 100000$。
算法讨论	对每一个结点定义一个局面函数 f，其定义方式为： ①若为叶子，则函数值为 0； ②若非叶子，其函数值为所有孩子的贡献之和。对于孩子 i，若连接它的是红边，令 a 为 $f_i + a > 1$ 的最小正整数，否则令 a 为 $f_i + a < -1$ 的最大负整数，其贡献即为 $\frac{f_i + a}{2^{\lvert a \rvert - 1}}$。 若根结点 f 值为 0，则后手必胜；若为正数，则第一个人必胜；否则第二个人必胜。求 f 值的过程我们直接按定义 DFS 求解，按二进制做高精度。由于每个结点函数值二进制为 1 的位数不超过其子树大小，我们对每个结点开一棵平衡树来维护其高精度二进制位，启发式合并即可。
时空复杂度	时间复杂度 $O(N\log^2 N)$；空间复杂度 O(NlogN)。

No. 63

试题编号	Codechef MAY 14
试题名称	Dynamic Trees and Queries
题目大意	给定一个下标从 0 开始的 N 个结点的有根树，根为 0 号点，每个点有一个 value。M 次操作，强制在线，包含以下类型： ①对给定下标的结点增加一个孩子，其下标为从未使用的下标中最小的一个； ②对给定下标的结点的子树中所有结点的 value 加上给定值；

	③对给定下标的结点，删除其子树中所有结点，不释放下标； ④对给定下标的结点，询问其子树中所有结点的 value 和。 数据范围：$1 \le N, M \le 50000$。
算法讨论	求出树的 DFS 序，将一个点拆成入栈点和出栈点。那么①操作即是在序列中合适位置加入连续的两个点，②操作是区间加，③操作是区间删除，④操作是询问区间和。这些很显然都可以用 Splay 来维护。
时空复杂度	时间复杂度 O(MlogN)；空间复杂度 O(N)。

No. 64

试题编号	Codechef MAY 14
试题名称	Sereja and Subsegment Increasings
题目大意	给定两个长度为 N 的序列 A、B，数字均在模 4 意义下。每次操作可以选择 A 序列的一连续段，将其中元素增加 1 后模 4。问将 A 转化为 B，最少要执行多少次操作。 数据范围：$1 \le N \le 100000$。
算法讨论	在模 4 意义下，将 A 减去 B（对应位相减）；然后在非模意义下，对 A 差分得到 C，问题转化为一初始全 0 数组，每次可以选择两个下标 i、j $(i < j)$，将第 i 位加 1、将第 j 位减 1，问最少多少次操作得到数组 C。 若非模意义，此时答案显然为数组 C 中正数的和。考虑模 4，也就是我们可以选择 i、j $(i < j)$，将第 i 位加上 4、第 j 位减去 4。由于 A 数组中每一位的取值均在[-3, 3]之间，对于[-1, 1]，变换后答案不会更优，因而我们顺序遍历 C，每遇到-2 或-3 就标记下来，下次遇到 2 或 3 时，优先看有没有-3，再看有没有-2，若有则将之前的数+4，当前的数-4，更新答案。
时空复杂度	时间复杂度 O(N)；空间复杂度 O(N)。

No. 65

试题编号	Codechef JUNE 14
试题名称	Two Companies
题目大意	给定一棵 N 个结点的树，和两个树上的链集合 A、B。集合中每一条路径都有一个权值。试在两个集合中分别选出一个子集，使得不存在两条被选出的且属于不同集合的路径相交，求最大化的权值和。 数据范围：$1 \le N \le 100000$，$\|A\|, \|B\| \le 700$。
算法讨论	判定任意两条路径是否相交后，问题转化为最大权闭合图问题，这可以用网络流解决。问题在于如何判断两条路径相交。 我们把一条路径拆成 LCA 指向两端点的两条路径，那么只需判定两条由孩子指向祖先的路径是否相交，这又等价于判断一个点在不在这样的端点内部，又可以转化为一个点是否是另一点祖先的判定，这依旧可以使用 LCA 来解决。
时空复杂度	时间复杂度 $O(\|A\|\|B\|\log N + (\|A\| + \|B\|)^2\|A\|\|B\|)$；空间复杂度 $O(N\log N + \|A\|\|B\|)$。

No. 66

试题编号	Codechef JUNE 14
试题名称	Sereja and Arcs
题目大意	从(1, 0)到(N, 0)均匀分布了 N 个点，每个点都有一个颜色。现在在第一象限内，对每个颜色相等的点对间画一条圆弧，其颜色与点颜色相同，圆心在 x 轴上。

	求有多少对不同颜色的圆弧相交。 数据范围：$1 \le N \le 10^5$。
算法讨论	考察两条圆弧，其对应数对颜色关系只有 ABAB、AABB、ABBA 三种，我们用所有圆弧对的数量减去 AABB 和 ABBA 的数量来得到答案。 AABB 很容易计算，我们只需要求一个前缀圆弧数量和一个后缀圆弧数量即可。注意要减去一对弧同色的情况。 对 ABBA，我们分两种情况考虑： ①两种颜色中有一种个数大于 $\sqrt{\frac{N}{\log N}}$，那么可以扫一遍数组，维护某种颜色的左（右）边有多少个这种颜色的数来计算答案； ②两种颜色个数都不多于上述阈值，这样产生的弧数量是有限的，可以用树状数组来考虑每一条弧的贡献来得到。注意要减去一对弧同色的情况。
时空复杂度	时间复杂度 $O(N\sqrt{N\log N})$；空间复杂度 O(N)。

No. 67

试题编号	Codechef JULY 14
试题名称	Game of Numbers
题目大意	给定两个长度为 N 的数组 A、B，每一轮要选出两个数对(i, j)和(p, q)满足 $A_i < B_j$、$A_p > B_q$、$\gcd(A_i, B_j, A_p, B_q) > 1$、(i, j)不在集合 S1 中、(p, q)不在集合 S2 中。然后分别将两个数对加入 S1、S2 中。问最多能进行多少轮。 数据范围：$1 \le N \le 400$。
算法讨论	把每一个(i, j)，满足 $\gcd(A_i, B_j) > 1$，根据 A_i、B_j 的大小关系把所有这样的数对分成二分图。对于图中不在同一部分的两个点，若其 gcd（四个数的 gcd）大于 1，就在之间连边，求最大流即可。 然而这样边数过多，需要优化。我们可以把所有出现过的质数建成点放在图的两部分之间，图的两个部分中的点均以其权值（两个数的 gcd）是否被这些质数整除来连边，这样做和之前是完全等价的，但却大大减少了边数。 对于点数，我们还可以通过把同一部中权值相等的点合并来减小。
时空复杂度	时间复杂度 $O(N^3\log^{3/2}N)$；空间复杂度 $O(N^2\log N)$。

No. 68

试题编号	Codechef JULY 14
试题名称	Sereja and Equality
题目大意	两个长度为 N 的数组 A、B 相似，当且仅当对于所有 i ($1 \le i \le N$)，满足 $C(A, A_i) = C(B, B_i)$。其中 C(X, x)表示满足 X[j] < x 的 j 的数目。 对于两个排列 P1、P2，定义函数 F(P1, P2)等于满足 P1[l...r]相似于 P2[l...r] ($1 \le l \le r \le N$)并且 P1[l...r]包含不超过 E 个逆序对的数对(l, r)的数目。 T 组数据，每组给定 N、E，对 P1、P2 取遍 N 个元素的所有排列，求 F(P1, P2)的总和是多少。 数据范围：$T \le 10000$，$N \le 500$，$E \le 1000000$。
算法讨论	令 f[i][j]为长度为 i 的排列中逆序对数不超过 j 的个数，显然有 f[i][j] = f[i - 1][j] - f[i - 1][j - i + 1]，我们预处理此二维数组 f。

	枚举相似子区间的长度，对于长度为 L 的子区间，共有 N - L + 1 个选择的方法，且无论该子区间如何，与其相似的合法选择共 $C_N^L \times F[L][E]$ 种，其它位置的种数则是(N - L)!。累计答案即可。
时空复杂度	时间复杂度 $O(N^3 + TN)$；空间复杂度 $O(N^3)$。

No. 69

试题编号	Codechef AUG 14
试题名称	Team Sigma and Fibonacci
题目大意	给定 N、M，令 $fibonacci_0 = 0$、$fibonacci_1 = 1$ 求： $(\sum_{x+y+z=N且x,y,z为非负整数} 6 \cdot x \cdot y \cdot z \cdot fibonacci_x \cdot fibonacci_y \cdot fibonacci_z) \bmod M$ 。 数据范围：$0 \le N \le 10^{18}$，$1 \le M \le 100000$。
算法讨论	一通化简以后，我们可以发现： $原式 = N\sum 6xyf_x f_y f_z - 2\sum 6x^2 yf_x f_y f_z$（f 就是 fibonacci）。 继而又有： ①$\sum 6xyf_x f_y f_z = N^2 \sum f_x f_y f_z - 3\sum x^2 f_x f_y f_z$； ②$\sum 6x^2 yf_x f_y f_z = N^3 \sum f_x f_y f_z - 3\sum x^3 f_x f_y f_z - N\sum 6xyf_x f_y f_z$。 现在我们的任务就转化为了求 $\sum x^i f_x f_y f_z$。f_x 在模 M 意义下循环，且循环节长度 P 是 O(M)的，那么对于模 P 相同的值，我们就没必要多次计算。在 x, y, z < p 时我们对 i = 1, 2, 3 分别计算 $\sum f_x f_y f_z$ 贡献的系数，将分别得到一个关于 t = (N - x - y - z) / P 的多项式，手推其通项后可以 O(1)计算。 现在我们仍需要枚举 x、y、z（模 P 意义下）。然而注意到 y、z 只出现在 f_y、f_z 中，所以我们无需枚举，只需要获得 $\sum_{y,z<p且x+y+z\equiv N(\bmod P)} f_y f_z$ 即可。 定义 $g_x = \sum_{i=0}^{x} f_i f_{x-i}$（可递推 O(P)求出），并用之得到 $h_x = \sum_{0\le i<p且0\le x-i<p} f_i f_{x-i}$。 有了 h 数组之后，上述要获得的式子也就非常容易计算了。
时空复杂度	时间复杂度 O(M)；空间复杂度 O(M)。

No. 70

试题编号	Codechef AUG 14
试题名称	Push the Flow!
题目大意	给出一个 N 个点、M 条边的简单无向连通图，每条边有容量，满足每个结点至多属于一个简单环，Q 次操作，或询问以 S 为源点、T 为汇点时的最大流（S 与 T 不相等），或是修改某条边的容量。 数据范围：$N \le 100000$，$M, Q \le 200000$。
算法讨论	由于最大流等于最小割，所求的即是使 S、T 不连通的最小割。那么或是断掉

	两点间所有简单路径都包含的任何一条边、或是断掉不同在两者间任意简单路径上但在一个环内的两条边。 我们考虑用线段树和树链剖分来解决。将每个环断成链后接起来用一棵线段树来维护，再将环缩点。在得到的树中，把某一树点 u 的权值，设为以 u 的最顶端结点为 S、u 的父树点的最顶端结点为 T 的最小割（在刚刚建出的线段树中查询）。 继而我们做树链剖分，对于一次询问(S, T)，我们求出其 LCA，分别考虑两条深度递减的路径，并特判（然后直接在线段树中查询）一些特殊的情况，然后重链直接在树链剖分结构内查询、轻边在一开始的线段树内查询，取最小值即可；对于修改，若修改的是非环边，直接在树剖结构内单点修改，否则我们先在一开始建出的线段树中修改，再修改树剖结构内该环对应的树点的重儿子的权值。
时空复杂度	时间复杂度 $O(M + Q\log^2 N)$；空间复杂度 $O(N + M)$。

No. 71

试题编号	Codechef SEPT 14
试题名称	Rectangle Query
题目大意	给定一个平面，Q 次操作，包含以下几种： ①插入一个矩形； ②删除插入的某一条矩形； ③给定一个矩形，询问当前有多少个矩形与这个矩形有公共点。 数据范围：$Q \le 100000$。
算法讨论	首先补集转化，问题变为查找有多少个矩形与这个矩形没有公共点。也即 x、y 两个维度中至少在一个维度上这两个矩形不相交。我们可以求出不相交的四种情况，但是由于这四种情况存在交集，还需要减去多算的部分。 其中不相交的四种情况，直接用树状数组即可简单计算。多算的部分也只有四种类型，我们把所有操作对时间分治，则每种都相当于求有关三维偏序关系的问题，我们在分治过程中加入树状数组来解决。
时空复杂度	时间复杂度 $O(Q\log^2 Q)$；空间复杂度 $O(Q)$。

No. 72

试题编号	Codechef SEPT 14
试题名称	Fibonacci Numbers on Tree
题目大意	给出一棵 N 个结点的树，结点有权值（初始为 0），Q 次操作，包含以下类型： ①对一条简单路径，按次序分别加 Fibonacci 数列的第 i 项（设第一项与第二项都为 1）； ②询问在以某结点为根时，另一个结点的子树的权值和； ③询问某条路径上的权值和； ④将树还原到某一次操作后的状态。 数据范围：$N, Q \le 100000$。
算法讨论	我们任选一点为根做树链剖分。对于修改操作，容易观察到链加 Fibonacci 数列相当于加两个公比不同的等比数列，因而我们在线段树区间上分别记两个方向所加的两个公比数列的首项是多少，这样就能支持线段树标记的合并了。 对于询问路径，树剖很容易支持；对于询问子树，若在树剖的有根树中我们

	的询问点不是假定根的祖先，那么答案就是询问点的子树权和；若非，答案为整棵树的权和减去询问点的孩子中是假定根的祖先的那棵子树的权和（或询问点就是假定根，则不需要减）。 由于还要恢复历史版本，所以我们还要对线段树可持久化。
时空复杂度	时间复杂度 $O(Q\log^2 N)$；空间复杂度 $O(Q\log^2 N)$。

No. 73

试题编号	Codechef OCT 14
试题名称	Children Trips
题目大意	给出一棵 N 个结点的树，树上每条边的权值为 1 或 2。Q 组询问，每组给出 u、v、d，有一个人从 u 走到 v，每天最多走 d 的长度且必须终止于结点，问总共要走多少天。 数据范围：$N, Q \le 100000$。
算法讨论	离线处理询问，对于 d 相同的我们一起处理。 其中 d 不小于 $N^{0.5}$ 的，我们的答案不会超过 $N^{0.5}$。所以我们把路径拆成深度递减的两段。利用倍增算每一天走多远，暴力计算两段各要走多少天（下取整），再考虑一下多出来部分的贡献。 若 d 小于 $N^{0.5}$，我们可以倍增预处理出每个点向祖先走 2 的次幂天能走到哪里，那么就相当于将前一种情况的暴力计算天数改为倍增。
时空复杂度	时间复杂度 $O((Q + N)N^{0.5}\log N)$；空间复杂度 $O(N\log N)$。

No. 74

试题编号	Codechef NOV 14
试题名称	Chef and Churu
题目大意	给出一个长度为 N 的数组和 N 个函数，第 i 个函数值为数组中以连续区间的和。共 Q 个操作，或是对数组的单点，或是询问一个连续区间的函数值的和。 数据范围：$N, Q \le 100000$。
算法讨论	把函数分成 $N^{0.5}$ 块，在每一块中预处理数组的每一位贡献的系数，并计算出一开始每一块的函数值和。对于单点修改，我们只需要进入每一块中按贡献系数更新即可；对于询问，我们对一块中的直接获得答案，对不够块的用树状数组每次获得一个函数的值。
时空复杂度	时间复杂度 $O(QN^{0.5}\log N)$；空间复杂度 $O(N^{1.5})$。

No. 75

试题编号	Codechef NOV 14
试题名称	Sereja and Order
题目大意	有 N 个程序，都需要在两台电脑上分别运行。其中第 i 个程序在第一台电脑上需要运行 A[i]秒、在第二台电脑上需运行 B[i]秒。一台电脑不能同时运行两个程序、一个程序不能同时在两台电脑上运行，最小化完成所有程序执行任务的时间，要求输出方案。 数据范围：$N \le 10000$。
算法讨论	答案的下界为 $\max(\sum_{i=1}^{N} A[i], \sum_{i=1}^{N} B[i], \max_{1\le i\le N}(A[i]+B[i]))$，易证下界是一定可以

	取到的。 对于外层 max 取第三种的情况，我们只需要把那个使 A[i] + B[i]最大的程序在两台电脑上错开时间，其余程序错开放即可；对于取前两种的情况（不妨设为第一种），打表发现合法的解是很多的，我们每次随机一个程序的序列，按顺序往两台电脑中加入（其中第二台电脑的插入位置要在尽量靠前的基础上保证错开在两台电脑上运行的时间），当发现第二台电脑的用时超过下界时，我们在第二台电脑上尽量把这之前已加入的程序往后排，把还没加入的程序往先放。至于第一台电脑则按顺序接起来即可。一旦发现合法解则停止随机。
时空复杂度	时间复杂度 O(TN)，T 是期望随机次数；空间复杂度 O(N)。

No. 76

试题编号	Codechef DEC 14
试题名称	Divide or die
题目大意	给定一个 N 度角，要求尺规作图（过已知点画直线、以已知点为圆心并以已知的两点距离为半径画圆，不超过 1000 次操作）将其 N 等分，输出方法以及最后中间 N - 1 条射线上的任意一点，或判断无解。 数据范围：N 为(0, 360)内的整数。
算法讨论	实现计算几何所需的各项函数，实现用尺规作图画 72 度角、画 60 度角、平分角和复制角。 若 N 是 3 的倍数则无解；否则我们可以先作一个 72 度角、再作一个 60 度角来获得 12 度角，二分再二分后获得 3 度角，复制之后一定能获得 N - 1 度或 N + 1 度角，这相当于我们得到了 1 度角，再将该角复制即可。
时空复杂度	时间复杂度 O(N)；空间复杂度 O(N)。

No. 77

试题编号	Codechef DEC 14
试题名称	Course Selection
题目大意	课业计划有 N 项课程，每项课程都需要在 M 个学期里的某一个完成（不一定每个学期都有所有课程）。一些课程有前置课程（可能不止一个，设共有 K 对前置关系），相同的课程在不同的学期里学习，获得的分数（满分 100）是不一样的。试最大化 N 项课程所得分数的平均值。 数据范围：$1 \le N, M \le 100$，$0 \le K \le 100$。
算法讨论	我们可以把问题转化为最小割。 将每一门课拆成 M 个点，在第 i - 1 个点和第 i 个点（视源点为第 0 个点、汇点为第 M + 1 个点）之间连边权为在第 i 个学期学习要失去的分数（满分与其做差，若第 i 学期没有该课程则为 inf）。对于 a 是 b 的前置的关系，则从 a 的每个点（含第 0 个点，即源点）向 b 的对应的下一个点连权值为 inf 的边。 如此之后，求最大流即可。
时空复杂度	时间复杂度 $O((N + K)N^2M^3)$；空间复杂度 $O((N + K)M)$。

No. 78

试题编号	Codechef JAN 15
试题名称	Ranka

题目大意	在一个 9 * 9 的棋盘上进行类似于围棋的游戏。一方执黑、一方执白，交替行棋，黑方先手。每一步中，玩家可选择在一空点放棋子（不使本方棋子无气）或放弃本轮，放置棋子后，需按围棋规则判断并进行提子。在避免同局再现的前提下，求一个含 N 步行动的游戏。 数据范围：$N \le 10000$。
算法讨论	把格子以行为第一关键字、列为第二关键字排序，倒序用黑色把除第一个格子外的棋盘填满，然后在第一个格子内下白子，再倒序用白色把除第二个格子外的棋盘填满，然后在第二个格子内下黑子……以此类推，直到进行了 N 步。 因为空任何一个格子都会贡献 160 步行动，所以能达到 81 * 160 = 12960（步），已经超过 10000 步。
时空复杂度	时间复杂度 O(N)；空间复杂度 O(1)。

No. 79

试题编号	Codechef JAN 15
试题名称	Xor Queries
题目大意	维护一个初始为空的整数序列，M 次操作，包括下列几种： ①在序列后加一个数字； ②给定一个数字，在给定区间中找到一个数字，最大化二者异或值； ③删除序列最后若干元素； ④统计给定区间中，比给定数字小的元素个数； ⑤在给定区间中，找给定排名的数。 数据范围：$M \le 500000$，数字均为[1, 500000]内的整数。
算法讨论	建立一棵可持久化 Trie，插入时把数字写成二进制串（对齐低位）。②④⑤操作均只需仿照可持久化权值线段树做区间 K 大值的方法，把对应的两棵 Trie 做差，从高位到低位在 Trie 上跑一遍。对于③操作，则直接返回历史版本。
时空复杂度	时间复杂度 O(MlogW)；空间复杂度 O(MlogW)。W 为最大的数字。

No. 80

试题编号	Codechef FEB 15
试题名称	Devu and Locks
题目大意	求有多少 N 位十进制数是 P 的倍数且每位之和小于等于 M，允许前导 0，答案对 998244353 取模。M 取遍 0 到 MM 的所有整数。 数据范围：$N \le 10^9$，存在 $P \le 50$、$MM \le 500$ 的数据和 $P \le 16$、$MM \le 15000$ 的数据。
算法讨论	设 f[x][y]为当前数模 P 为 x，位数和为 y 的方案数。 考虑每一位的贡献，对于 10^i（i 为第几位，从 0 开始从低位数）模 P 相等的每一位我们可以一起算（找 N 模 P 过程中出现的环即可优化）。假设当前考虑模 P 为 w 的这些位且这些位有 k 个，那么我们设生成函数 g(x)，i 次方项表示位数和为 i 的方案数，那么有： $g(x)=\left(\sum_{i=0}^{9}x^i\right)^k$，这可以用倍增 FFT 来计算。 继而容易推出 f[i * w mod P][i] = $[x^i]$g(x)，我们共得到了 P 个这样的 f 数组，

	只需将其两两合并即得答案。合并过程相当于一个二维卷积，由于第一维较小，故我们可以建立第一维维数个的数组，对应数组间 FFT 做多项式乘法，中途可以一直使用点值表达式计算，最后再 IDFT 回去。
时空复杂度	时间复杂度 $O(P^2MM(P + \log MM))$；空间复杂度 $O(MMP)$。

No. 81

试题编号	Codechef FEB 15
试题名称	Payton numbers
题目大意	定义三元组(a, b, c)的乘法运算，其中 c = 11 或者 c = 24。如果令(a_1, b_1, c_1)和(a_2, b_2, c_2)相乘，则步骤如下： ①令 $s = (a_1a_2 + b_1b_2 + c_1c_2) + (a_1b_2 + b_1a_2) + (c_1 + c_2)$，$t = [s / 2] + 16(c_1 + c_2) - c_1c_2$； ②再令 $A = (t - 2(a_1b_2 + b_1a_2) - (a_1c_2 - c_1a_2) + 33(a_1 + a_2) + (b_1b_2 - a_1a_2))$；$B = (t - 5(a_1b_2 + b_1a_2) - (c_1b_2 + b_1c_2) + 33(b_1 + b_2) + (2b_1b_2 + 4a_1a_2))$； ③若 s 是偶数，结果就是(A - 540, B - 540, 24)，否则结果就是(A - 533, B - 533, 11)。 定义单位元 A 是对于任何 B 都满足 A * B = B 的三元组，定义 zeroA 是对任何 B 都满足 A * B = A 的三元组。定义一个三元组是素数当且仅当这个三元组不能表示成两个非零非单位元的三元组的乘积。 给定一个三元组，判断它是否为素数。 数据范围：$T \le 10^4$，$\|a\|, \|b\| \le 10^7$。
算法讨论	令ω为满足方程$\omega^2 = \omega - 3$ 的解，那么对于每一个三元组(a, b, c)都有到域 Z[ω]的映射 f(a, b, c) = (33 - 2a - c) + (b - a)ω。问题转化为判断域 Z[ω]下的数 a + bω 是否为素数。 定义共轭(a + bω)' = (a + b - bω)，那么令 Nx = xx'，就有如下结论： ①若 x 不是整数，那么 x 是质数当且仅当 Nx 是质数； ②若 x 是整数，那么 x 是质数当且仅当 x 是质数，并且|x| = 2 或|x| ≠ 11 且-11 在模 x 域中无二次剩余。 利用上述结论，就可以直接用勒让德符号和 Miller-Rabin 算法来判断。
时空复杂度	时间复杂度 $O(\log\|a\|)$；空间复杂度 $O(1)$。

No. 82

试题编号	Codechef MARCH 15
试题名称	Counting on a Tree
题目大意	给定一个含 N 个结点无根树，有边权。给定 Q 组操作，每组将某一条边的权值进行修改，每次要求输出满足经过边权的 gcd 为 1 的简单路径的条数（含初始局面）。 数据范围：$N \le 100000$，$Q \le 100$，任意时刻边权为不超过 10^6 的正整数。
算法讨论	令 f(i)为树上边权为 i 的倍数的路径的条数，那么答案为$\sum_{i=1}^{10^6} f(i)\cdot\mu(i)$。 现在我们求 f(i)。我们只需要考虑所有边权为 i 的倍数的边，因而我们建出一张点不变但只有这些边的新图。如果没有修改，可以通过并查集维护连通分量来计算这些边在图中形成的路径条数。对于本题来说，修改的次数有限，

	因而我们可以把和修改无关的边先用并查集维护，对于每次修改（含初始状态）我们考虑在这种状态中出现在该图的边后再删除，用并查集启发式合并来支持还原并查集。
时空复杂度	时间复杂度 $O(S(N + Q^2)\log N)$；空间复杂度 $O(S(N + Q^2))$。S 为权值平均因子数。

No. 83

试题编号	Codechef MARCH 15
试题名称	Random Number Generator
题目大意	给定一个 K 阶线性递推式每一项的系数 C_i（C_i 为推出序列某一项时往前数 i 项的系数）和由其构成的 K 阶线性递推序列前 K 项，求序列第 N 项。 数据范围：$N \le 10^{18}$，$K \le 30000$。
算法讨论	该线性递推式对应的转移矩阵的特征多项式为 $f(x) = x^k - \sum_{i=1}^{k} C_k x^{k-1}$。令 g(x) = $x^N \bmod f(x)$，答案就是 $\sum_{i=1}^{k} [x^{i-1}]g(x) \times C_i$。我们用多项式求逆实现多项式除法，用多项式除法实现多项式取模，求出 g(x)即可。
时空复杂度	时间复杂度 $O(K\log K\log N)$；空间复杂度 $O(K)$。

No. 84

试题编号	Codechef APRIL 15
试题名称	Black-white Board Game
题目大意	有一个 N * N 全为白色的矩阵。游戏开始前，将在每一行选定一个连续列染成黑色。两名玩家轮流行动，每次都要给出一个 1 到 N 的排列 P，其中先手给出的排列的逆序对是偶数、后手反之，并且对任意 i 满足矩阵中(i, P_i)对应的格子为黑色。所给出的排列不得重复，给不出排列者告负。问双方均采取最优策略时，谁会获胜，或判断平局。 数据范围：$N \le 100000$。
算法讨论	将黑色格子视为 1、白色格子视为 0，该矩阵的行列式即为偶数的合法排列数减去奇数的合法排列数。我们考虑用消元法和拉普拉斯展开来计算矩阵的行列式。 由于为黑色的格子在每一行中均为连续区间，我们可以顺序考虑每一列，找出黑色格子从该列开始且结束的最早的那一行，用这一行去消其它这一列为黑的行；此后再对这一行展开（事实上相当于删掉这一行、这一列）后计算代数余子式。 消元过程可以使用可并堆进行优化，对于每一列可并堆（小根堆）中维护从该列开始为黑色块的所有行（以结束块的列为关键字）。消元过程即是将当前列的堆并入消掉的连续块的下一列对应的堆中。由于求代数余子式，我们还要维护新矩阵标号。
时空复杂度	时间复杂度 $O(N\log N)$；空间复杂度 $O(N)$。

No. 85

试题编号	Codechef APRIL 15
试题名称	Little Party
题目大意	给定 M 个不同的长度为 N 的 01 串，一个基子集可以用一个数列 S 和一个长度为|S|的 01 串 t 表示，一个基子集能覆盖的 01 串 A 满足对于任意的 $1 \le i \le \|S\|$，都有 $t_i = A_{S_i}$。现在要用一些基子集，使得所有给定的串都可以被这些基子集覆盖且没有给定的串都没有被这些基子集覆盖。试最小化基子集大小的和。 数据范围：$M \le 1000$，$N \le 5$。
算法讨论	该问题转化为最小带权集合覆盖问题，由于是 NPC 问题，我们只能搜索。我们先枚举所有的基子集，判断其是否可行并存储。 若一个基子集能被另一个基子集完全包含，那么这个基子集是无效的，加上这条限制后发现此时最多有 32 个基子集。在搜索时使用位运算、可行性剪枝、最优化剪枝以及记忆化，就可以在时限内出解。
时空复杂度	时间复杂度 $O(2^S)$；空间复杂度 $O(2^S + NM)$。S 为合法且有效的基子集个数。

No. 86

试题编号	Codechef MAY 15
试题名称	Chef and Balanced Strings
题目大意	一个字符串被称为平衡的当且仅当它的字符可以被分到两个完全相等的多重集合。给出一个长度为 N 的由小写字母构成的字符串 P，并给定 Q 组询问，每组给出 L、R、Type，其中前两个数表示 P 的一个子串的起止下标，计算该子串的子串中，所有平衡的字符串的长度的 Type 次幂的和，强制在线。 数据范围：$N, Q \le 100000$，$Type \in \{0, 1, 2\}$。
算法讨论	对每个字母出现次数的奇偶性状态压缩，并维护 P 的前缀状态 S[i]。则子串 P[L...R]平衡当且仅当 S[L - 1] = S[R]，我们对 S[i]离散化后可使其范围变为 O(N)。 继而我们可以将整个字符串分成 $N^{0.5}$ 块，通过维护每一种状态出现下标的个数、下标的和、下标的平方和，我们能够预处理出从任一点开始到某一块终止、从某一块开始到任一点终止的答案。 那么对于一个询问，我们将答案写成起始点到最后一个整块的答案 + 第一个整块到终止点的答案 + 起始点所在块与终止点所在块的答案 - 整块间的答案，还没计算的部分仿照预处理时的方法即可。
时空复杂度	时间复杂度 $O(N^{1.5})$；空间复杂度 $O(N^{1.5})$。

No. 87

试题编号	Codechef MAY 15
试题名称	Counting on a directed graph
题目大意	给定一个 N 个点、M 条边的有向图，试统计无序对(X, Y)的个数。其中(X, Y)满足存在一条从点 1 到点 X 的路径和一条从点 1 到点 Y 的路径，且两条路径除了点 1 以外没有公共点。 数据范围：$1 \le N \le 100000$，$1 \le M \le 500000$。
算法讨论	一个点对(X, Y)是合法的当且仅当从点 1 出发，点 X 和点 Y 没有除了点 1 以外的公共必经点。 我们以点 1 为根求出这张有向图的 dominator tree，那么点 X 和点 Y 是合法的

	条件就转化为它们的 LCA 是点 1。这可以对整棵树 DFS 一遍求出根结点的孩子结点的子树大小直接计算。
时空复杂度	时间复杂度 O(Nα(N))；空间复杂度 O(N)。

No. 88

试题编号	Codechef JUNE 15
试题名称	Chefbook
题目大意	给定 M 个限制 a_i、b_i、W_i、L_i、R_i，要求设定 N 个非负整数 X_i 和 N 个非负整数 Y_i，满足对于每一个限制都有 $L_i \le X_{a_i} - Y_{b_i} + W_i \le R_i$。试最小化上述中间式子的和，并输出一组合法方案。 数据范围：$N \le 100$，$M \le N^2$。
算法讨论	可以把限制写成 $x_i - x_j \le K$ 的形式，其中 x 是需要设定的 2n 个变量。对于每一个限制，在 i 到 j 之间连上一条流量无穷、费用为 K 的边。若变量 X 在限制中出现了 f 遍，就从源点到它连一条流量为 f 费用为 0 的边，对于变量 Y 则连向汇。在这张由对偶原理得到的流量图上做最小费用最大流，费用即为最小化的值。 对于图中的某一条边，若其流量不为 0，说明这一限制被刚好达到，我们再新建一图加入不等式限制，在新图中做差分约束即得合法方案。
时空复杂度	时间复杂度 $O(N^2M)$；空间复杂度 O(M)。

No. 89

试题编号	Codechef JULY 15
试题名称	Easy Exam
题目大意	有一个 K 面等概率的骰子，每个面上数字分别是 1 到 K。给定参数 L（不超过 K）和 F。将该骰子掷 N 次，令 cnt[i]为掷出数字 i 的次数，求 $\prod_{i=1}^{L} cnt[i]^F$ 的期望。 数据范围：$0 < N, K \le 10^9$，$0 < LF \le 20000$，$0 < F \le 1000$。
算法讨论	设 $event_{i,j}$ 为第 j 次投是否投到 i，显然有 cnt[i]为 $event_{i,j}$ 对固定 i、任意 j 求和。答案要求的则是这一求和式的次幂的积。我们将整个答案式展开，若一项中存在不同的 a、b、c 使得 cnt[a][c]和 cnt[b][c]的指数都大于 0，那么这一项的贡献为 0，否则若这一项出现了 size 个不同变量，易证其贡献为 K^{-size}。 考虑计算有 j 个不同变量的贡献不为 0 的系数。令 w[i][j]为展开$(\Sigma x_k)^i$ 出现 j 个不同变量的系数，这可以用 w[i][j] = w[i - 1][j - 1] + w[i - 1][j] * j 来递推得到。那么我们做 i 次求和式次幂的乘积，有 j 个不同变量的系数之和可以通过 w[f]数组合并来获得（要考虑组合数）。合并过程实际上就是一个卷积，我们用 FFT 实现多项式快速幂算出系数后直接计算答案。
时空复杂度	时间复杂度 O(LFlogLFlogL)；空间复杂度 O(LF)。

No. 90

试题编号	Codechef JULY 15
试题名称	A game on a graph

题目大意	在一个 N 个点、M 条边、无自环的无向图上两个人进行游戏。后手选择一个起始点，将硬币放在该点，二者轮流把硬币沿着一条边移到另一个点，硬币不能重复到达同一个点，无法操作的玩家失败。称一个点为胜利点，当且仅当选择该点为起点后后手必胜（假设双方均按最优策略）。求有多少个胜利点。 数据范围：$N \le 2000$，$M \le 300000$。
算法讨论	一个点是必胜的当且仅当存在一个原图的最大匹配使得这个点在这个匹配中是孤立点。故我们先用带花树求一遍最大匹配。此时和当前的任意孤立点间存在一条长度为偶数的增广路径的所有点都可能在一次增广之后变成孤立点，于是我们可以从每一个孤立点开始再做一次增广，把所有距离为偶数或者花中的结点标记为必胜。最后统计被标记的点数。
时空复杂度	时间复杂度 O(NM)；空间复杂度 O(M)。

以下题目类型为 Challenge：

No. 91

试题编号	Codechef SEPT 12
试题名称	Simultaneous Nim
题目大意	将 N 含堆石子的石子序列 A 分成若干（记为 K）组，同时在 K 组石子中进行 Nim 游戏，当玩家均采取最优策略时，要求 K 个游戏均为后手必胜。试求一个满足要求、且 K 最大的划分。 数据范围：$10 \le N \le 1000$，$1 \le A[i] \le 2^{60} - 1$，所有 A[i]异或值为 0。
算法讨论	问题即为将 N 个数划分成最多的组数，使每组数的 xor 均为 0。我们考虑随机化，按一定顺序往临时集合中加数，同时维护高斯消元的数组，一旦当前的数已经可以异或成 0，就把这些数分到一个新集合中并删去，重复上述过程直到序列为空。高斯消元的数组可以用 bitset 优化。
时空复杂度	时间复杂度 O(60TNK)，T 由调参得到；空间复杂度 O(60N)。

No. 92

试题编号	Codechef DEC 14
试题名称	Kali and Devtas
题目大意	给定平面上 N 个点的坐标，求一个生成树，最小化 C[i]的最大值。 C 值的计算：对于每个点，设其在生成树中相邻的最远点的距离为 R，那么离该点距离为 R 以内的点的 C 值全部加 1。 数据范围：$3 \le N \le 400$。
算法讨论	不难看出，生成树中的边小一点，从总体上来说会使 C[i]被加的次数减小，事实上对于本题，直接求最小生成树答案已经非常优秀了。更进一步的，我们可以通过随机交换两条边位置等方法，使用爬山或退火来获得更优的解。
时空复杂度	时间复杂度 $O(N^2 \log N)$；空间复杂度 $O(N^2)$。

No. 93

试题编号	Codechef FEB 13
试题名称	Efficient Painting
题目大意	一个正方形由 N * N 的小正方形格子组成，每个格子可能为黑或白，初始全白。

	每次可以将一个矩形内的颜色涂黑、涂白或反转，最小化获得给定局面的步数，输出最小化的方案。 数据范围：$10 \le N \le 50$。
算法讨论	我们考虑只用反转操作来完成这个问题，这时从给定局面还原白色与原问题等价。对于一个格子(i, j)，我们记以其为左上角的 2 * 2 的小正方形中黑色的个数 B[i][j]，那么一次反转之后，只有所选矩形的左上角格子的左上格子、右上角格子的正上格子、左下角格子的正左格子、右下角格子的 B 值的奇偶性会发生变化，而且全白状态当且仅当每个格子的 B 值都为偶数（事实上就是 0）。 因此，我们按照上述四个点 B 为奇数的个数来贪心，优先选择对应的这四个点的 B 值为奇数的个数多的矩形，对于含相同的 B 值为奇数的特殊点的矩形，则随机选择。多次重复取最优解。
时空复杂度	时间复杂度 $O(TLN^3)$，T 由调参得到；空间复杂度 $O(L + N^2)$。L 为期望步数。

No. 94

试题编号	Codechef MARCH 14
试题名称	Sereja and Sorting 2
题目大意	一个长度为 N 的数组 A，可以通过每次翻转数组的一个区间来使其变为升序，一次翻转的代价是区间长度，最小化(翻转次数 + 翻转代价和 / N)并输出方案。 数据范围：$1 \le N \le 10000$，不同的数的个数不超过 1050。
算法讨论	假设我们目标是将[L, R]排序（初始是[1, N]）。 特判下列情况：若 A[L]已经在它该放的位置上，我们将问题变成排序[L + 1, R]；若 A[L]应该在 A[R]的位置、A[R]应该在 A[L]的位置，我们翻转这一整个大区间，并缩减问题为[L + 1, R - 1]；若当前考虑区间的最小值和次小值的重复次数均为 1，且次小值出现在最小值之前，我们两次翻转使其到区间开头，并使 L 自增 2。 注意到本题中不同的数的个数是有限的，这意味着相同的数的个数很多。我们考虑一次将当前考虑区间中和最小的数相同且个数不超过阈值的数一起翻转到子区间开头，具体方法就是从后向前，逐步将这些数翻转到一起。 不难看出，上述方法兼顾了权值和与步数两个方面，适当调整阈值就能获得很优秀的解。
时空复杂度	时间复杂度 $O(N^2)$；空间复杂度 $O(N)$。

No. 95

试题编号	Codechef APRIL 12
试题名称	Similar Graphs
题目大意	给出两个 N 个点的图的邻接矩阵（无重边、自环），要求对两张图的点进行重标号，最大化(2 * 公共边对数 / 两张图总边数)并输出方案。 数据范围：$30 \le N \le 75$。
算法讨论	不难看出我们只需要对一张图进行重标号，考虑随机化。每次选择两个结点对应的标号并尝试交换之，仿照模拟退火，若权值更大则接受改变、权值小时也按照其小了多少一定概率接受。每一轮做 $O(N^2)$次，做多轮取最优解。计算权值的过程可以使用 bitset 优化。

时空复杂度	时间复杂度 $O(TN^2)$，T 由调参得到；空间复杂度 $O(N^2)$。

No. 96

试题编号	Codechef AUG 13
试题名称	Deleting numbers
题目大意	有一长度为 N 的数组 A，每次可以删除下标为等差数列且值均相等的一系列数（若按公差往下标增大方向推得的下一个数也满足条件，则必须加入），然后重标号。最小化删除所有数所需的次数，输出方案。 数据范围：$1 \le N \le 100000$。
算法讨论	用链表维护值相等的数所在原始序列的位置，用树状数组 O(logN)的二分来实现动态下标。每次考虑序列最末尾的数，若它之前没有和它相等的数，直接删除；否则以它到其前一个数的下标差为公差，贪心删除能删掉的数。
时空复杂度	时间复杂度 O(NlogN)；空间复杂度 O(N)。

No. 97

试题编号	Codechef AUG 11
试题名称	Complex Spanning Tree
题目大意	在一个 N * N 的网格图上求生成树，其中每条边的权值都是一个复数。要求最大化生成树权值和的模，输出选择的边。 数据范围：$N \le 100$。
算法讨论	假如最优解答案的极角为α，那么我们把所有边的权值用其向量形式到该极角对应向量的投影大小来代替，直接做最大生成树，一定能求得我们所求的最优解。因而我们使用撒点爬山法来找这个极角，在$[0, 2\pi]$的区间均匀撒若干个点作为初始极角，每次用当前极角转过一个角度得到的新极角给所有的边赋值，然后做最大生成树，若比当前解优则更新极角，减小转动的步长后重复直到步长值足够小。对撒的不同点的答案再求最大值。
时空复杂度	时间复杂度 $O(N^2\log D)$，D 为最大步长与最小步长的比值；空间复杂度 $O(N^2)$。

No. 98

试题编号	Codechef OCT 12
试题名称	Maximum Sub-rectangle in Matrix
题目大意	给出一个 H * W 的整数矩阵 A，选出一个子矩阵，最大化子矩阵元素的和，要求输出选择的行、列。 这里的子矩阵不要求连续，等价于选出的若干行和若干列在所在处相交的元素。 数据范围：$200 \le H, W \le 300$。
算法讨论	随机选择一些行，容易求出在确定这些行的情况下，选哪些列答案最优；再用选出来的列去求出在确定这些列的情况下，选哪些行答案最优。如此反复迭代，直到行、列均不再发生变化。重复以上过程，选最优解。
时空复杂度	时间复杂度 O(TKHW)，T 由调参得到，K 为期望迭代次数；空间复杂度 O(HW)。

No. 99

试题编号	Codechef JUNE 13
试题名称	To challenge or not

题目大意	给出一个长度为 N 的数组 B，要求从中选出数组 A，使得 A 中任意三个数不构成等差数列，最大化 A 的长度，输出方案。 数据范围：$N \le 100000$，B 数组中元素为不超过 100000 的非负整数。
算法讨论	要求等价于 A 数组中不存在一个数，它乘 2 以后是其它两个不同的数的和。我们将 B 数组排序，按升序考虑 B 中元素 B[cur]是否放在 A 中，则当前 A 中任意满足 $i < j$ 的(i, j)，有 $A[j] + A[j] - A[i] \neq B[cur]$，这只需要我们在往 A 中插入元素时，暴力枚举 A 中已存在的元素，将按上式左式形成的那个数值标记为不可进入数组。下一次有新的 B[cur]考虑是否加入时直接 O(1)判断即可。但是这样贪心不一定是最优解，我们对可以加入 A 的 B[cur]进行一次随机，使其有概率不加入到 A 中。反复以上过程，利用时间戳优化，取最优解。
时空复杂度	时间复杂度 $O(T(L^2 + N))$，T 由调参得到，L 为 A 数组期望长度；空间复杂度 O(N)。

No. 100

试题编号	Codechef JUNE 12
试题名称	Closest Points
题目大意	给出 N 个三维空间中的点，Q 组询问，每次给定一个点左边，询问 N 个点中与其距离最近的点的下标。 数据范围：$N, Q \le 50000$，坐标绝对值不超过 10^9。
算法讨论	此题是一个经典的 KDTree 求最近点的问题。我们只需要在 KD 树上每个结点记录当前结点及其子结点中最大、最小的各维坐标，查询时就可以利用这些信息进行剪枝。KDTree 的期望单次查询复杂度是 O(logN)的，但在本题中由于数据特殊的构造方法，几乎每次询问都达到了 O(N)，因而这需要我们进行优化才能在时限内获得答案。 首先，建树时不再按照 0~2 的顺序循环各维度，对于某一个尚未确定是哪一个点的结点以及其所对应的备选序列，考虑当前序列中三维方差最大的那一维，并以这一维作为划分的依据，这样做不会影响建树的时间复杂度，但将大大提升我们剪枝的效率。此外在查询时利用估价函数，优先选择估价最大的那棵子树递归，同时对查询函数调用的次数设置一个阈值，当超过后直接跳出并返回当前找的的答案。做完这些步骤后，将可以既准确又快速地获得答案。
时空复杂度	时间复杂度 O(NlogN + QS)，S 为设定的阈值；空间复杂度 O(N)。