试题泛做

by 吉如一

January 18, 2016

题 1

试题编号	Codechef MAY 15
试题名称	Chef and Balanced Strings

题目大意	算法讨论
一个字符串是平衡的当且仅当它的每一个字符都出现了偶数次，一个字符串的 $type$ 权值是它的所有平衡子串长度的 $type$ 次方和。现在给定一个长度为 n 的由小写字符构成的字符串，有 Q 组询问，每一次询问由 $L, R, type$ 描述，表示询问这个字符串的一个子串的 $type$ 权值是多少。 数据范围 $n, q \leq 10^5, type \in \{0, 1, 2\}$	我们对每一个小写字母分配一个不同的 2 的幂次表示权值，然后令 x_i 表示字符串前 i 个字符权值的异或值，那么子串 $[L, R]$ 是平衡的当且仅当 $x_{L-1} = x_R$，我们对 x 离散化就可以让它在 $O(n)$ 的范围内。 接着对下标分块，块的大小和块数均为 $O(\sqrt{n})$。我们可以对于每一个 i 和 k 预处理出前 i 个块中所有 x_j 为 k 的下标 j 的数目、和以及平方和，并可以预处理出任意两个块之间的答案。 对于每一组询问，我们把它分成三段：一个块的后缀，一段连续的块，一个块的前缀。于是我们可以把答案分成三部分统计：连续块内的答案，多余段之间的答案以及两端分别在多余段和连续块中的答案。 对于第一部分我们已经预处理出来了，而对于第二部分因为多余段的总长度是 $O(\sqrt{n})$ 的，所以可以直接开一个数组扫一遍得到答案，对于第三个部分我们可以枚举在多余段中的端点并利用预处理的块内下标数目、和、平方和得到答案。这儿举一个例子：假设我们枚举到的下标为 j，在询问的第一段中，且第二段里权值和 j 相等的下标个数是 A，下标和是 B，平方和为 C，那么它对 $type = 0$ 的贡献就是 A，对 $type = 1$ 的贡献就是 $B - A \times j$，对 $type = 2$ 的贡献就是 $C + A \times j^2 - 2B \times j$（之前的预处理也是使用这个方法）。把这三部分的贡献累加起来就是答案了。

时空复杂度	时间和空间复杂度都是 $O(n\sqrt{n})$。

题 2

试题编号	Codechef MAY 15
试题名称	Counting on a directed graph
题目大意	算法讨论
给定一张 n 个点 m 条边的有向图，一个点对 (X, Y) 是合法的当且仅当存在一条从 1 到 X 的路径，一条从 1 到 Y 的路径满足它们除了 1 号点以外没有任何公共点。统计合法的无序对 (X, Y) 的个数。 数据范围 $n \le 10^5, m \le 5 \times 10^5$	可以发现一个点对 (X, Y) 是合法的当且仅当从 1 出发，点 X 和点 Y 没有除了点 1 以外的公共必经点。 所以我们可以以 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了它们的 LCA 是点 1。于是就可以对这一棵树 DFS 一遍求出每一个子树的大小，然后扫一遍点 1 的儿子得到答案。
时空复杂度	时间复杂度 $O(n\alpha_n)$，空间复杂度 $O(n)$。

题 3

试题编号	Codechef APRIL 15
试题名称	Black-white Board Game
题目大意	算法讨论
一个 $n \times n$ 的 01 矩阵，它的第 i 行的第 L_i 到 R_i 个格子为 1，其余格子都是 0。 现在有两个人在玩游戏，每一轮游戏他们同时报出一个还没有被报出过的长度为 n 的排列 P，排列 P 必须满足第 i 行第 P_i 个格子是 1，且第一个人报出的排列的逆序对个数必须是奇数，第二个报出的必须是偶数。 如果有一轮一个人找不到可能的排列而另一个人能找到那么就算能找到的人赢，如果两个人同时找不到则算平局，你需要判断一个游戏结果。 数据范围 $n \le 10^5$	我们把逆序对为偶数的排列的权值设为 1，为奇数排列的权值设为 -1，然后对所有满足第 i 行第 P_i 列的格子时 1 的排列求出它们权值的和。那么可以发现当权值和为正数时是第二个人赢，是负数时是第一个人赢，为 0 时是平局。 根据行列式的定义，可以发现求和得到的值就是初始矩阵的行列式，于是只需要求出这个矩阵行列式的值就好了。但是直接消元的时间复杂度是 $O(n^3)$ 的，所以必须要用数据结构优化。 考虑建 n 棵左偏树，对于第 i 行，我们最开始把 R_i 给插入第 L_i 棵左偏树中。然后我们从小到大遍历这 n 棵左偏树，如果当前左偏树为空，说明消元后有对角线上有一个值为 0，行列式必定为 0；否则我们找到这棵（第 L 棵）左偏树中最小的 R_i，在删除它之后把这颗左偏树与第 $R_i + 1$ 棵左偏树合并，这就相当于选出这一行与所有第 L 个格子为 1 的行消元，可以保证消元后每一行的 1 依然是连续一段。这样就可以快速完成消元了。
时空复杂度	时间复杂度 $O(n \log n)$，空间复杂度 $O(n)$。

题 4

试题编号	Codechef MARCH 15
试题名称	Random Number Generator

题目大意	算法讨论
给定一个数列的递推式：$A_i = (A_{i-1} \times C_1 + A_{i-2} \times C_2 + ... + A_{i-k} \times C_k) \mod 104857601 (i > k)$，现在已知 A_i 和 $C_i (1 \le i \le k)$，求 A_n。 数据范围 $n \le 10^{18}, k \le 3 \times 10^4$	这是一个线性递推式，显然是可以用矩乘来优化的，但是直接矩乘的复杂度是 $O(k^3 \log n)$。可以发现求出这个矩阵特征多项式为 $f(x) = x^k - \sum_{i=1}^{k} C_i x^{i-1}$，所以可以用特征多项式来优化矩阵乘法，这是一个经典问题，令 $g(x) = x^n \mod f(x)$，那么答案就是 $\sum_{i=1}^{k} [x^{i-1}] g(x) \times C_i$。为了求得 $g(x)$，我们可以使用 FFT 进行快速幂的运算并在用多项式除法取模，时间限制很紧要做许多常数优化。

时空复杂度	时间复杂度 $O(k \log k \log n)$，空间复杂度 $O(k)$。

题 5

试题编号	Codechef FEB 15
试题名称	Devu and Locks

题目大意	算法讨论
对于所有的 $0 \le M \le MM$，求出满足个位数字之和不超过 M 且是 P 的倍数的 N 位数（可以有前导 0）。 两类数据范围： $n \le 10^9, p \le 50, MM \le 500$ $n \le 10^9, p \le 16, MM \le 15000$	令 $dp_{i,j,k}$ 为各位数字之和为 k 且模 P 得 j 的 i 位整数个数（可以有前导 0），我们可以考虑分治。 假设已经知道了所有的 $dp_{i,j,k}$，那么显然有 $dp_{2i,j,k} = \sum dp_{i,a,b} \times dp_{i,(j-a) \mod P, k-b}$ 如果直接暴力的话复杂度是 $O(P^2 MM^2 \log n)$，但是可以使用 FFT 优化。即把数组看成 $dp_{i,j}(x)$，那么就有 $dp_{2i,j}(x) = \sum dp_{i,a}(x) \times dp_{i,(j-a) \mod P}(x)$ 于是我们可以先 DFT 所有的 $dp_{i,j}(x)$，然后把所有的和累积起来，最后再把 $dp_{2i,j}(x)$ 给 IDFT 回去得到答案。这样就能在时限内得到答案了。

时空复杂度	时间复杂度 $O(P^2 MM \log n + PMM \log MM \log n)$，空间复杂度 $O(MMP)$。

题 6

试题编号	Codechef NOV 14
试题名称	Chef and Churu

题目大意	算法讨论
给定一个长度为 n 的数组和 n 个函数，第 i 个函数值为数组中一个连续区间的和。接下来有 m 个操作，每个操作是接下来两种之一：单点修改数组内的值和询问一个连续区间的函数值的和。 数据范围 $n, m \le 10^5$	对操作序列分块，每 $O(\sqrt{m})$ 次操作重新计算一次所有函数的权值，这可以通过预处理前缀和来做到 $O(n) - O(1)$，所以这部分的复杂度是 $O(n\sqrt{m})$ 的。我们可以再对数组 x 分块，可以先预处理出对于第 i 块第 j 个位置的贡献，这样对于每一次修改可以 $O(\sqrt{n})$ 更新每一块所有函数的权值和。对于一次询问，把它分成两部分，一部分是若干个连续整块，这个部分的和由刚才的修改已经可以 $O(\sqrt{n})$ 的得到。至于多余的部分，我们只有在得到当前数组的前缀和的情况下才能 $O(1)$ 得到一个函数的答案，所以最后要处理的只有区间加，单点求值。 我们可以对下标分块，每一次对于区间中的整块就打一个标记就好了，对于多余的部分暴力加，这样这个操作就变成了 $O(\sqrt{n}) - O(1)$。

时空复杂度	时间复杂度 $O(n\sqrt{n})$，空间复杂度 $O(n\sqrt{n})$。

题 7

试题编号	Codechef SEPT 15
试题名称	Rectangle Query
题目大意	算法讨论
在一个二维平面中，你需要支持以下三种操作： 1. 插入一个矩形。 2. 删除一个矩形。 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 $Q \le 10^5$	先对整个操作序列 CDQ 分治，问题就变成了插入一些矩形，接着有若干次询问，每一次询问有多少个给定矩形与询问矩形有交点。 可以把每一个矩形 $(x1, y1), (x2, y2)$ 用线段 $y1 - y2$ 和区间 $[x1, x2]$ 表示，与每一个矩形相交的矩形等价于线段相交的矩形减去区间右端点比询问矩形左端点小的线段相交的矩形数目再减去区间左端点比询问矩形右端点大的线段相交的矩形数目。所以我们可以分这三部分分别统计。 每一部分可以把每一个矩形拆分成线段，然后对线段上端点和下端点分别维护一个树状数组，按照 x 轴坐标顺序分别插入询问。
时空复杂度	时间复杂度 $O(Q \log^2 Q)$，空间复杂度 $O(Q)$。

题 8

试题编号	Codechef SEPT 15
试题名称	Fibonacci Numbers on Tree
题目大意	算法讨论
给定一棵 n 个节点的树，你要支持以下操作： 1. 给 u 到 v 路径上的第 i 个点加上 fib_i，其中 fib_i 代表斐波那契数列第 i 项。 2. 询问以 x 为根时 y 的子树和。 3. 询问 x 到 y 路径上所有节点的权值和。 4. 让整棵树回到第 i 次操作后的状态。 答案可能很大，请对 $10^9 + 9$ 取模后输出。 数据范围 $n, Q \le 10^5$	我们对这棵树树链剖分，并在树链剖分的时候记录每一个点访问到的次序 dfs_i 以及它子树中 dfs_i 的最大值 r_i。 对于第一种操作，$fib_i = \frac{1}{\sqrt{5}}[(\frac{1+\sqrt{5}}{2})^i - (\frac{1-\sqrt{5}}{2})^i]$，其中在 $\sqrt{5}$ 在模 $10^9 + 9$ 是存在一个整数的，我们可以用这个整数代表式子中的全部 $\sqrt{5}$，可以发现问题转化成了区间加两种公比的等比数列，询问区间和。可以对每一种公比 p 维护一个标记 f_i，表示在当前区间加上首项为 f_i 公比为 p 的等比数列，这个标记显然是可以合并的。所以可以用传统的树链剖分来维护这个操作。 对于第二种操作和第三种操作，第二种对应着线段树中 $O(\log n)$ 个区间，第三种对应着线段树中 $O(1)$ 个区间。所以可以直接对每一个区间维护区间和，用线段树来查询。 对于第四种操作，我们可以把树链剖分的线段树可持久化，即全部用可持久化线段树来维护，这样就可以回到历史版本了。
时空复杂度	时间复杂度 $O(Q \log^2 n)$，空间复杂度 $O(Q \log^2 n)$。

题 9

试题编号	Codechef APRIL 15
试题名称	Chef and Tree Game

题目大意	算法讨论
有一棵 n 个节点的树，树上每一条边的颜色都为红色或者蓝色。现在两个人轮流进行操作，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，若干轮之后不能操作的人算输。 如果两个人都使用最优策略，问第一个人先手时、第二个人先手时分别是谁赢得游戏。 数据范围 $n \le 10^5$	对每一个节点定义一个局面函数 f，可以使用递推的方式来得到：叶子的函数值为 0。每一个节点的函数值为所有孩子的贡献之和，对于一个孩子 i，如果连接它的是红边，那么令 a 为 $f_i + a > 1$ 的最小正整数，它的贡献就是 $\frac{f_i+a}{2^{a-1}}$；如果连接的是蓝边，那么令 a 为 $f_i - a < -1$ 的最小正整数，它的贡献就是 $\frac{f_i-a}{2^{a-1}}$。 如果根节点的函数值是为 0，那么谁后手谁赢；如果是整数，那么无论如何都是第一个人赢；否则无论如何都是第二个赢。 至于求函数值，可以用高精度来实现，可以发现一个大小为 i 的子树根节点函数值的位数不超过为 i，所以可以对每一个子树开一棵平衡树来维护每一位，然后用启发式合并的方式来实现两个高精度数的加。

时空复杂度	时间复杂度 $O(n \log^2 n)$，空间复杂度 $O(n \log n)$。

题 10

试题编号	Codechef APRIL 15
试题名称	Make It Zero 3

题目大意	算法讨论
现在有 m 个数组和一个整数 P，第 i 个数组用 A_i 表示，且由四个整数 F_i, C_i, D_i, L_i 生成，生成方式如下： 1. 数组 A_i 的长度为 L_i 2.$A_{i,0} = F_i$ 3.$A_{i,j} = (A_{i,j-1} \times C_i + D_i) \mod P$ 接着，把这 n 个数组依次连接得到一个数组 B（即可以理解为 $B = \sum_{i=1}^{m} A_i$）。然后你需要对这个数组进行若干次操作，每一次操作你可以任意选择数组中的一个区间 $[L, R]$ 和一个整数 k 并把所有下标在 $[L, R]$ 中的数 A_i 变成 $(A_i + k) \mod P$。 问至少需要多少次操作才能把数组中的所有数都变成 0。 数据范围 $m \le 100$,$P \le 10$,$L_i \le 10^{15}$	把数组差分，即定义 $x_i = B_i - B_{i-1}$，问题可以转化为把数组 x 划分为尽可能多的子序列使得每一个子序列的和是 P 的倍数。因为对于每一个长度为 n 子序列我们都可以用 $n-1$ 次操作把它变成 0，所以最终的操作次数就是 $n+1-k$，其中 k 是子序列个数。 可以发现对于所有 $i + j = P$ 的数对 i, j，肯定是尽可能把 i 和 j 凑到一组，所以最后要考虑的最多只有 5 种数字，而且其中必定有一种数字至多有一个。我们可以预处理出所有可能在答案中的子序列，这个子序列一定满足它不存在一个子集的和为 P 的倍数，所以子序列的数目是有限的，打表可以发现至多有 30 种。 我们可以先随便选出一种初始解，然后考虑优化这个解，每一个优化一定是若干个子序列分裂成更多的子序列。打表后可以发现不能被其他优化线性表出的优化只有 130 个左右，且每一个优化的每一个数的出现次数不超过 12。所以可以用 DP 求出所有优化然后暴力搜索至最优解。 至于 x 数组中每一个数字的出现次数可以找每一个子数组的循环节得到。

时空复杂度	时间复杂度 $O(12^4 tot + \text{玄学})$，其中 tot 是可能出现在最优解中的子序列个数，空间复杂度 $O(m)$。

题 11

试题编号	Codechef JULY 15
试题名称	Easy exam

题目大意	算法讨论
你有一个 K 面的骰子，投到每一面的概率都是完全相同的，现在你需要投 n 次，设投完之后数字 i 出现了 a_i 次，试求 $\sum_{i=1}^{L} a_i^F$ 的期望值。 数据范围：$n, K \le 10^9$,$1 \le L \le K$,$0 \le L \times F \le 5 \times 10^5$,$1 \le F \le 1000$	设 $x_{i,j}$ 为第 j 次投是否投到 i，如果投到就是 1 否则就是 0。那么显然有 $a_i = \sum_{j=1}^{n} x_{i,j}$，答案要求的就是 $\prod_{i=1}^{L}(\sum_{j=1}^{n} x_{i,j})^F$。我们把这个式子展开，显然每一项是独立的，我们可以对每一项分开来考虑，如果一项中存在不同的 a, b 和 c 使得 $x_{a,c}$ 和 $x_{b,c}$ 的指数都大于 0，那么这一项的贡献就是 0，否则设这一项出现了 $size$ 个不同的变量，那么它的期望值就是 $\frac{1}{K^{size}}$。 考虑计算有 i 个不同变量的贡献不为 0 的的系数。令 $w_{i,j}$ 为式子 $(\sum x_k)^i$ 的展开中出现了 j 个不同变量的系数之和（假设每一个变量都是本质相同的），那么显然有递推式 $w_{i,j} = w_{i-1,j-1} + w_{i-1,j} \times j$。令 $dp_{n,m}$ 为 $\prod_{i=1}^{n}(\sum x_{i,j})^F$ 的展开中出现了 j 个不同变量系数之和的方案数，那么就有 $$dp_{n,m} = dp_{n-1,j} \times w_{F,m-j} \times \binom{m}{j}$$ 这个递推是可以用 FFT+ 快速幂优化的。这样就得到了每一项的系数，然后最后累加答案就好了。

时空复杂度	时间复杂度 $O(LF \log LF \log L)$，空间复杂度 $O(LF)$。

题 12

试题编号	Codechef AUG 13
试题名称	Prime Distance On Tree

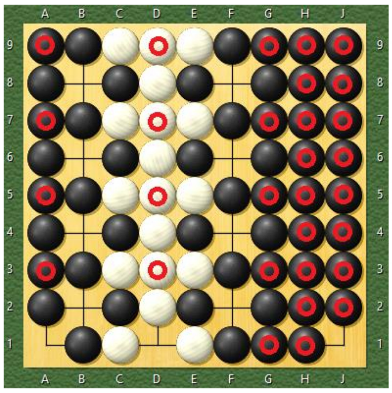
题目大意	算法讨论
给定一棵 n 个节点的树，现在在树上随机的选一条路径 $(u, v)(u < v)$，问路径长度是质数的概率是多少。 数据范围：$n \le 50000$	可以直接统计出长度为 i 的路径有多少条。我们可以考虑对整棵树进行点分治，每一次合并两个部分的时候，先分别统计出两个部分长度为 i 的路径的长度是多少（假设分别为 A_i 和 B_i），那么对长度无哦为 k 的路径条数的贡献就是 $\sum_{i=0}^{k} A_i \times B_{k-i}$，这个可以用 FFT 来优化。

时空复杂度	时间复杂度 $O(n \log^2 n)$，空间复杂度 $O(n)$。

题 13

试题编号	Codechef MARCH 15
试题名称	Counting on a Tree
题目大意	算法讨论
给定一颗 n 个节点带边权的树，接下来有 Q 次操作，每一次操作修改一条边的权值。所有操作前以及每次操作后输出这棵树中所有权值为 1 的路径条数，路径的权值定义为路径上所有边权值的最大公约数。 数据范围：$n \le 10^5, Q \le 100$，边权不超过 $Z = 10^6$	令 f_i 为树上权值为 i 的倍数的路径条数，那么答案就是 $\sum_{i=1}^{Z} f_i \times \mu_i$。所以只要对于所有的 $i \in [1, Z]$ 求出 f_i 就能得到答案了。 假设现在在求 f_i，那么我们只需要考虑所有边权为 i 的倍数的边，如果没有修改，就可以通过并查集来求出只考虑这些边的图中的路径条数。如果有修改那么就只需要先把和修改无关的边插到并查集中，然后对于每一次修改把修改后依然在这张图中的边插进去，然后再复原就好了。因为存在复原操作，所以要使用启发式合并的并查集。
时空复杂度	时间复杂度 $O(Z + 2^x(n + Q^2)\log n)$，空间复杂度 $O(2^x n)$，其中 x 为 $[1, Z]$ 中最大质因子个数。

题 14

试题编号	Codechef JAN 15
试题名称	Ranka
题目大意	算法讨论
你需要在一个 9×9 的围棋棋盘上，构造出一个 n 步且不存在重复局面的围棋下法（双方都由你控制）。 数据范围：$n \le 10^4$	 如图，在 9×9 的棋盘上最多可以构造出 8 个结，这 8 个结的状态有 2^8 个，通过下围棋的方法遍历这些位置可以通过格雷码来实现，因为格雷码也是相邻两个状态不同的二进制位不超过一个，第 i 个格雷码等于 i 异或 $(i >> 1)$。遍历一遍这些状态最多可以消耗 1.5×2^8 步。 因为除了打结需要的点以外还有 30 个点（图中标记出来的点）可以利用，每放下一个棋子，我们都可以把打结的所有状态都遍历一遍，这样就可以产生大约 $30 \times 1.5 \times 2^8$ 步，已经超过 10^4 了。
时空复杂度	时间复杂度 $O(n)$，空间复杂度 $O(1)$。

题 15

试题编号	Codechef JAN 15
试题名称	Xor Queries
题目大意	算法讨论
最开始你有一个空序列，接下来有 n 个操作，每一个操作是下列操作中的一种： 1. 在序列末尾插入一个数。2. 删去序列末尾的 k 个数。3. 询问区间 k 小值。4. 询问区间与 x 异或值最大的数。5. 询问区间不比 x 大的数的个数 数据范围：$n \le 5 \times 10^5$，数字范围不超过 5×10^5	我们可以仿照主席树，使用函数式 Trie 来解决这个问题，只需要对区间的每一个前缀维护一棵 Trie 树就可以解决每一个询问操作了。因为每一次至多插入一个数，所以只需要在插入这个数的时候，新建所有修改过的点，其余点沿用前驱树的下标就好了。
时空复杂度	时间复杂度 $O(n \log n)$，空间复杂度 $O(n \log n)$。

题 16

试题编号	Codechef DEC 14
试题名称	Course Selection
题目大意	算法讨论
有 n 门课和 m 个学期，每一门课都必须选一个学期学习（每一个学期都有可能有课不开课），第 i 门课在第 j 个学期学习可以获得 $A_{i,j}$ 的分数（如果不开课 $A_{i,j} = -1$）。同时课程中存在 K 个前置关系，每一个关系用 A_i 和 B_i 描述，即第 A_i 门课必须在第 B_i 门课之前学习。问学完这 n 门课的分数之和最大是多少。 数据范围：$n, m, K \le 100$	可以把问题转化为最小割，我们把每一门课拆成 $m+1$ 个点 $p_{i,0}$ 到 $p_{i,m}$，在 $p_{i,j-1}$ 到 $p_{i,j}$ 之间连一条边表示在第 j 个学期学习要损失的分数，即学习第 i 门课能获得的最大分数减去 $A_{i,j}$（注意要先把 -1 变成 $-\infty$）。最后再在 $s->p_{i,0}$ 和 $p_{i,m}->t$ 之间连上无穷大的边。对于限制关系 a, b，我们可以在 $p_{a,i}$ 与 $p_{b,i+1}$ 之间连上无穷大的边。这样这张图的最小割就是学完所有课最少要损失的分数，用所有课的最大分数之和减去它就是答案了。
时空复杂度	时间复杂度 $O(maxflow(n, nm))$，空间复杂度 $O(nm)$。

题 17

试题编号	Codechef OCT 14
试题名称	Children Trips
题目大意	算法讨论
给定一棵 n 个点的树，树上每一条边的权值都是 1 或 2。接下来有 Q 组询问，每组询问有一个人从 u_i 走到 v_i，他每天最多可以走 d_i 的长度，且每一天的终止点必须是树上的顶点，问他需要走几天。 数据范围：$n, Q \le 10^5$	对于 $d_i \ge \sqrt{n}$ 的询问，答案是 $O(\sqrt{n})$ 的，所以可以先倍增预处理出第 i 个点往父亲走 2^j 的距离能走到哪里，然后把过程给拆分成两个点分别到 LCA 的两段，于是就可以分别模拟两段分别要走多少天，对于多出来的一段判一下需要多少天走完 对于 $d_i \le \sqrt{n}$ 的询问，我们把所有 d_i 相同的询问一起处理，对于每一个点可以先预处理出从这个点出发向上走一天能走到哪里，然后倍增求出向上走 2^i 天能走到哪里，然后就可以 $O(\log n)$ 处理每一次询问了。
时空复杂度	时间复杂度 $O(n\sqrt{n}\log n)$，空间复杂度 $O(n \log n)$。

题 18

试题编号	Codechef AUG 13
试题名称	Music & Lyrics
题目大意	**算法讨论**
给定 n 个字符串 S 和 m 个字符串 T，每一个字符串都只包含大小写字母数字以及 -，询问每一个 S 串在 T 中一共出现了多少次。 数据范围：$n \le 500, \lvert S\rvert \le 5000, m \le 100, \lvert T\rvert \le 50000$	可以先把所有的 S 串给插入一个 AC 自动机中，然后对每一个串 T 都在 AC 自动机上匹配一遍，对于每一个匹配到的点都把它的权值 +1。那么每一个 S 串的答案就是 fail 树上它所在点的子树和。
时空复杂度	时间复杂度 $O(n\lvert S\rvert + m\lvert T\rvert)$，空间复杂度 $O(n\lvert S\rvert)$。

题 19

试题编号	Codechef AUG 14
试题名称	Team Sigma and Fibonacci
题目大意	**算法讨论**
给定 n, m，求 $\sum(6xyz \times fib_x \times fib_y \times fib_z \times [x+y+z=n]) \mod m$ 数据范围：$n \le 10^{18}, m \le 10^5$	可以推出这个数列的母函数为 $f(x) = 6g(x)x^3(x^2+1)^3$，其中 $[x^n]g(x) = \frac{1}{3125}[25\binom{n+5}{5}(fib_{n+6} + 2fib_{n+5}) + 150\binom{n+4}{4}fib_{n+5} + 5\binom{7}{2}\binom{n+3}{3}(fib_{n+4} + 2fib_{n+3}) + 5\binom{8}{3}\binom{n+2}{2}fib_{n+3} + \binom{9}{4}(n+1)(fib_{n+2} + 2fib_{n+1}) + \binom{10}{5}fib_{n+1}]$ 所以就可以直接计算了，求 fib_i 的时候可以用矩乘直接搞，求组合数的时候可以暴力约分。
时空复杂度	时间复杂度 $O(\log n)$，空间复杂度 $O(1)$。

题 20

试题编号	Codechef JULY 14
试题名称	Game of Numbers
题目大意	**算法讨论**
给定两个长度为 n 数组 A, B，每一轮你要选出两个数对 (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 $S2$ 中。然后把 (i,j) 加入集合 $S1$，(p,q) 加入集合 $S2$，问最多能进行多少轮游戏。 数据范围：$n, m \le 400$	我们把每一个 $\gcd(i,j) > 1$ 的数对都拿出来，可以发现根据 A_i 和 B_j 的大小关系可以把所有数对分成两个部分。对于图中两个点，如果满足这两个点的 gcd 大于 1，就在它们之间连一条边，那么这张图的最大流就是答案。但是这样边数是 $O(n^4)$ 的，所以要考虑优化连边。 对于图中每一个出现过的质数 p，我们新建一个点，并对于所有 gcd 是 p 的倍数的点向这个点连入（出）一条边，可以发现这张图和之前的建图是完全等价的。这样就可以在时间限制内跑出来了。
时空复杂度	时间复杂度 $O(maxflow(n^2, n^2\log n))$，空间复杂度 $O(n^2\log n)$。

题 21

试题编号	Codechef JULY 14
试题名称	Sereja and Equality
题目大意	算法讨论
两个长度为 n 的数组 A,B 是相似的当且仅当对于任意的 i 都有 $CA(A_i) = CB(B_i)$，其中 $CX(i)$ 表示数组 X 中满足 $X_j \le i$ 的个数。 关于两个长度为 n 的数组 A,B 的函数 $F(A,B)$ 的值等于满足 $A[l...r]$ 和 $B[l...r]$ 相似且 $A[l...r]$ 的逆序对个数不超过 E 的数对 (l,r) 的个数。 t 组询问 n,E，求 $\sum F(P1,P2)$ 对一个质数取模的值，其中 $P1,P2$ 都是 $1-n$ 的排列。 数据范围：$n \le 500$,$E \le 10^6$,$t \le 10^4$	令 $dp_{i,j}$ 为长度为 i 的排列中逆序对个数不超过 j 的个数，那么显然有递推式 $dp_{i,j} = dp_{i-1,j} - dp_{i-1,j-i+1}$，所以我们可以用 $O(n^3)$ 的递推得到数组 dp。 考虑枚举相似子区间的长度，对于长度为 l 的子区间，它有 $n-l+1$ 个位置，它在每一个排列中的取值都有 $\binom{n}{l} \times dp_{l,E}$ 种，且每一个排列的其他位置都有 $(n-l)!$ 种排列方法。所以直接累计答案即可。
时空复杂度	时间复杂度 $O(n^3 + tn)$，空间复杂度 $O(n^3)$。

题 22

试题编号	Codechef JUNE 14
试题名称	Two Companies
题目大意	算法讨论
给定一棵 n 个节点的树以及两个树上链的集合 A 和 B，集合中每一条路径都有一个权值，你需要在两个集合中分别选出一个集合使得不存在两条被选出的且属于不同集合的路径相交。问选出来路径权值和最大是多少。 数据范围：$n \le 10^5$,$\lvert A\rvert,\lvert B\rvert \le 700$	显然这是一个二分图最大权匹配的问题，只需要判断任意两条路径是否相交，就能用经典的最大流算法来解决。 可以把每一条路径拆分变成两个部分，即两个端点朝向它们的 LCA 的两段路径，问题就转化为了判断两条由孩子指向祖先的路径是否相交，这个问题等价于判断一个端点在不在一条这样的路径的内部，即快速判断一个点在不在一颗子树内。这个可以用 DFS 序来解决，即预处理出每一个节点子树中的 DFS 序范围，然后判断一下它们是否相互包含即可。
时空复杂度	时间复杂度 $O(n\log n + maxflow(\lvert A\rvert + \lvert B\rvert, \lvert A\rvert\lvert B\rvert))$，空间复杂度 $O(n + \lvert A\rvert\lvert B\rvert)$。

题 23

试题编号	Codechef FEB 13
试题名称	Observing the Tree
题目大意	算法讨论
给定一棵 n 个节点的树，要支持以下三种操作：1. 路径加等差数列。2. 询问路径和。3. 恢复到第 i 次修改后的情况。 数据范围：$n,m \le 10^5$	首先可以对这颗树进行树链剖分，然后用线段树来维护，那么显然是可以打标记的，对线段树上每一个节点用 A,B 表示对这个区间加上首项为 A 公差为 B 的等差数列，这是可以合并的。因为有第三种操作，所以要把所有线段树可持久化。
时空复杂度	时间复杂度 $O(m\log^2 n)$，空间复杂度 $O(m\log^2 n)$。

题 24

<table>
<tr><td>试题编号</td><td colspan="2">Codechef NOV 12</td></tr>
<tr><td>试题名称</td><td colspan="2">Arithmetic Progressions</td></tr>
<tr><td colspan="2">题目大意</td><td>算法讨论</td></tr>
<tr><td colspan="2">给定一个长度为 n 数组 A，询问有多少对 $(i,j,k)(1\le i<j<k\le n)$ 满足 $A_i+A_k=2A_j$
数据范围：$n\le 10^5, A_i\le 3\times 10^4$</td><td>可以对序列分块，每一块大小为 $O(\sqrt{n\log n})$，对 (i,j,k) 所在块的情况进行讨论：
1.i 和 k 中至少有一个和 j 在同一个块中，可以先预处理出前（后）i 个块中权值等于 j 的数字个数，然后就可以直接枚举 (i,j,k) 中的两个的值然后进行统计，这一部分的复杂度是 $O(n\sqrt{n\log n})$ 的。
2. 三个均在不同块中，枚举 i 所在块，那么只需要求出 A_i+A_k 等于每一个值的对数即可，这可以用 FFT 来实现，这一部分的复杂度也是 $O(n\sqrt{n\log n})$ 的。</td></tr>
<tr><td>时空复杂度</td><td colspan="2">时间复杂度 $O(n\sqrt{n\log n})$，空间复杂度 $O(n\sqrt{n})$。</td></tr>
</table>

题 25

<table>
<tr><td>试题编号</td><td colspan="2">Codechef DEC 13</td></tr>
<tr><td>试题名称</td><td colspan="2">Query on a tree VI</td></tr>
<tr><td colspan="2">题目大意</td><td>算法讨论</td></tr>
<tr><td colspan="2">给定一棵 n 个节点的树，每一个节点都是黑色或者白色，你需要支持这两种操作：1. 修改一个点的颜色。2. 如果只保留树上连接两个相同颜色点的边，询问一个点所在联通块的大小。
数据范围：$n,m\le 10^5$</td><td>我们可以对黑色和白色两种颜色分别维护一个树链剖分，对于每一个节点维护假定它是黑色（白色）且只考虑它的子树时它所在黑色（白色）联通块大小。这可以直接用传统的维护子树和的方法维护，修改颜色的时候只需要在两条树链剖分上更新修改节点到根的路径就行了，询问的时候就找到距离它最远的路径上都是和它同色点的祖先，然后询问树链剖分上的单点值即可。因为要支持的操作只有区间加，所以用树状数组就行了。</td></tr>
<tr><td>时空复杂度</td><td colspan="2">时间复杂度 $O(n\log^2 n)$，空间复杂度 $O(n)$。</td></tr>
</table>

题 26

<table>
<tr><td>试题编号</td><td colspan="2">Codechef MAY 14</td></tr>
<tr><td>试题名称</td><td colspan="2">Dynamic Trees and Queries Solved</td></tr>
<tr><td colspan="2">题目大意</td><td>算法讨论</td></tr>
<tr><td colspan="2">给定一棵 n 个节点的带点权的树，接下来进行 m 次操作，操作有以下四种：1. 加入一个给定点权的叶子。2. 删除一个子树。3. 给一个子树中的所有节点的权值加上一个值。4. 询问一个子树中的权值和。（树根为 1 号节点）强制在线。
数据范围：$n,m\le 10^5$</td><td>因为子树加用 LCT 比较难处理，所以可以考虑转化为单点操作。每一个节点的权值记为它所在的子树中所有点加上了的值，那么询问节点 i 时的答案就是 $size_i\sum_j w_j+\sum_k w_k\times size_k$，其中 k 是 i 的孩子，j 是 i 的祖先。维护这个可以转化为路径修改询问单点和的问题，可以直接用 LCT 来维护。标记是路径的 $size$ 值以及 $\sum_k w_k\times size_k$ 值的修改，然后要维护路径的权值和以及单点的 $size$ 和 $\sum_k w_k\times size_k$。因为不存在换根操作所以就不用维护 rev 标记了。</td></tr>
<tr><td>时空复杂度</td><td colspan="2">时间复杂度 $O(n\log n)$，空间复杂度 $O(n)$。</td></tr>
</table>

题 27

试题编号	Codechef JUNE 14
试题名称	Sereja and Arcs

题目大意	算法讨论
给定一个长度为 n 的序列 A，对于每一对 $(i,j)(i<j)$，如果满足 $A_i=A_j$，那么就在坐标系中画上一个颜色为 A_i 的以 $(i,0),(j,0)$ 为直径的圆。问有多少对颜色不同的圆存在交点。 数据范围：$n,A_i\le 10^5$	因为数对的关系只可能为 $ABAB,AABB,ABBA$ 三种情况，可以考虑用总数减去 $AABB$ 和 $ABBA$ 的数目来得到答案。其中 $AABB$ 很好得到，只需要求出右端点为 $1-i$ 的圆的个数以及左端点为 i 的圆的个数即可。 对于 $ABBA$，分两种情况考虑：1. 两种颜色中有一种的数的个数大于 $\sqrt{\frac{n}{\log n}}$，那么就可以通过求出另一种颜色的每一个位置的左侧（右侧）有多少个这种颜色的数，然后把每一对的贡献拆开来累加得到答案。这一部分的时间复杂度是 $O(n\sqrt{n\log n})$。2. 两部分的个数都不多于 $\sqrt{\frac{n}{\log n}}$，可以发现所有满足这个条件的颜色产生的弧的个数是 $n\sqrt{\frac{n}{\log n}}$ 的，所以可以直接使用二维数点的方法来统计这一部分的贡献，这一部分的复杂度也是 $O(n\sqrt{n\log n})$ 的。 注意要扣除相同颜色的贡献。

时空复杂度	时间复杂度 $O(n\sqrt{n\log n})$，空间复杂度 $O(n)$。

题 28

试题编号	Codechef MAY 14
试题名称	Sereja and Subsegment Increasings

题目大意	算法讨论
给定两个长度为 n 的序列 A,B，满足 $0\le A_i,B_i<4$，每一次可以选出一个区间 $[l,r](1\le l\le r\le n)$，把 A 数组的区间 $[l,r]$ 在模 4 的意义下加一，问最少进行多少次操作后能把数组 A 变成数组 B。 数据范围：$n\le 10^5$	首先把 A_i 在模 4 意义下减去 B_i，接着令 $C_i=A_{i+1}-A_i$，这样问题就转化为了最开始有一个全为 0 的数组，每次可以选择两个数 $i,j(0\le i<j\le n)$，把第 i 位在模意义下加一，第 j 位减一，问最少多少次操作变成数组 C。 如果不是模意义下的操作，答案显然是 $\sum_{i=0}^{n}\max(0,C_i)$。于是问题就变成了可以对于 $i<j$ 把 C_i 加上 4，把 C_j 减去 4，最小化数组 C 中正数的和。 因为 C_i 只可能为 -3 到 3 且最后 C 的和为 0，所以对于 $-1\le C_i\le 1$，修改并不会使更优。而对于剩下的数可以用贪心的方法，即每找到 2 或者 3，都判断一下当前有没有 -3（如果没有就找 -2），如果有就把当前位置 -4，那个位置 +4 纳入答案。

时空复杂度	时间复杂度 $O(n)$，空间复杂度 $O(n)$。

题 29

试题编号	Codechef MARCH 14
试题名称	The Street

题目大意	算法讨论
给定一个长度为 n 的数列 A 和 B，开始数组 A 为 0，数组 B 为负无穷大。接下来有 m 次操作：1. 对于数组 A 区间加一个等差数列。2. 对于数组 B 区间对一个等差数列取 max。3. 询问 $A_i + B_i$ 的值。 数据范围：$n \leq 10^9, m \leq 3 \times 10^5$	操作 1 是经典问题，直接对线段树每一个节点记录这个区间被加上了首项为 A，公差为 B 的等差数列，显然标记是可以合并的。 对于操作 2，可以对线段树的每一个节点记录 x 和 y，表示这个区间对首项为 x，公差为 y 的等差序列取 max。在打标记的时候，一个节点可能同时对两个等差数列 x, y 和 x', y' 取 max。如果其中一个等差数列在这个区间中一直比较大，那么就保留这个等差数列。否则两个等差数列的优势区间一定是一个前缀和互补的后缀。其中一定有一个区间是只被包含在一个子树中的，这时我们可以把优势区间涵盖两个子树的等差数列标记在当前节点上，另一个等差数列则下传到子树中。

时空复杂度	时间复杂度 $O(m \log^2 n)$，空间复杂度 $O(m \log n)$。

题 30

试题编号	Codechef MARCH 14
试题名称	Chef and Graph Querie

题目大意	算法讨论
给定一张 n 个点 m 条边的无向图，并给出 Q 个询问 $[L_i, R_i]$，对于每个询问你需要输出如果只保留编号在区间 $[L_i, R_i]$ 中的边的话图中有多少个联通块。 数据范围：$n, m, Q \leq 2 \times 10^5$	离线，我们从小到大枚举询问的右端点，把每一条边的权值标记为它的编号，假如当前右端点为 i，那么求出只包含边 $[1, i]$ 时图中的最大生成森林，对于询问 $[l, i]$，如果此时最大生成森林中有 k 条权值大于等于 l 的边，那么答案就是 $n - l$。 所以需要维护动态加边的最大生成森林，这是 LCT 经典问题。至于查询边权大于等于 l 的边的数目可以用树状数组来实现。

时空复杂度	时间复杂度 $O(n \log n)$，空间复杂度 $O(n)$。

题 31

试题编号	Codechef FEB 14
试题名称	Graph Challenge

题目大意	算法讨论
给定一个 n 个点 m 条边的 DFS 序，每一个点的标号为它的 DFS 序。保证所有节点都能从第一个点到达。一个节点 x 对 y 来说是好的当且仅当 $x < y$ 且存在一条 x 到 y 的路径使得中间节点编号都大于 y。一个节点 x 对 y 来说是最好的当且仅当它是所有对 y 的好节点中编号最小的。 给定 Q 个询问，每个询问问对于一个节点，它是多少个节点的最好的节点。 数据范围：$1 \leq n, Q \leq 10^5, n - 1 \leq m \leq 2 \times 10^5$	根据定义，可以发现一个节点的最好的节点就是 dominator tree 算法中的半必经点。所以只需要以 $1 - n$ 这个 DFS 序运行 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 $O(n)$。

题 32

试题编号	Codechef FEB 14
试题名称	Count on a Treap

题目大意	算法讨论
有一个初始为空的 max-treap，现在进行了 n 次操作：1. 插入一个给定权值和重量的节点。2. 删除一个节点。3. 询问两个节点之间的路径长度。 保证权值和重量两两不同。 数据范围：$n \le 2 \times 10^5$	treap 的 DFS 序就是权值大小顺序，可以发现两点的 LCA 就是它们对应区间中重量最大的节点。所以可以在 $O(\log n)$ 的时间内求出两点之间的 LCA，接下来就只需要计算一个节点的深度就可以了。 观察性质可以发现一个序列中第 y 个节点是第 x 个节点的祖先当且仅当区间 $[x, y]([y, x])$ 中不存在重量比 y 大的节点。尝试使用线段树来维护一个节点左侧（右侧）的祖先个数。令函数 $getl(a, b)$ 为线段树第 a 个节点，当前最大值是 b 时询问这个区间范围内的祖先个数，这个只要维护区间最大值就能递归到一侧的子树中。所以每一次更新的复杂度是 $O(\log n)$，用线段树维护的时间复杂度是 $O(\log^2 n)$。（只要事先离散化就可以用线段树来支持插入和删除了）

时空复杂度	时间复杂度 $O(n \log^2 n)$，空间复杂度 $O(n)$。

题 33

试题编号	Codechef JAN 14
试题名称	Counting D-sets

题目大意	算法讨论
在 n 维空间中，两个点的距离定义为每一维坐标差的绝对值的最大值，一个点集的直径定义为距离最远的两个点的距离，两个点集是相同的当且仅当它们可以通过平移得到。询问 n 维空间中直径等于 D 的点集个数。 数据范围：$n \le 10^3, D \le 10^9$	可以极短直径小于等于 D 的点集个数，那么答案就是 ans_D 减去 ans_{D-1}。为了避免两个点集是相同的，可以限定每一维坐标必须有一个值为 0。那么所有点必须限定在一个边长为 D 的 n 维立方体中。令 w_i 为第 $1-i$ 维的坐标不存在为 0 的值，其他坐标都存在为 0 的值的方案数，这可以用简单的容斥算出来。

时空复杂度	时间复杂度 $O(n^2)$，空间复杂度 $O(n^2)$。

题 34

试题编号	Codechef JAN 14
试题名称	Counting The Important Pairs

题目大意	算法讨论
给定一张 n 个点 m 条边的无向图，询问有多少种删掉恰好两条边的方案使得这张图不连通。 数据范围：$n \le 10^5, m \le 3 \times 10^9$	可以先求出这张图的一棵 DFS 树，我们对每一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，如果删掉它可以让这张图不连通，那么一定存在一个子集使得这个子集的边权的异或和为 0。如果边权在 $[0, 2^{64})$ 范围内随机那么出错的概率就可以忽略不计。 因为这题中只删除两条边，所以一个方案是满足条件的当且仅当两条边边权相同或者有一条边边权为 0。直接求出所有边权然后排序扫一遍就行了。

时空复杂度	时间复杂度 $O(m \log m)$，空间复杂度 $O(m)$。

题 35

试题编号	Codechef DEC 13
试题名称	Petya and Sequence

题目大意	算法讨论
给定一个长度为 n 的整数序列 A，问是否存在一个不全为 0 的整数序列 B 满足对于所有的 $0 \le j < n$ 都有 $\sum_{i=0}^{n-1} A_i \times B_{(i+j) \bmod n} = 0$。 数据范围：$T \le 100, n \le 3 \times 10^4$	问题等价于判断矩阵 $X(X_{i,j} = A_{i+j \bmod n})$ 是否满秩。如果 $\gcd(f(x), x^n - 1)$ 的次数为 d，那么这个矩阵的秩就是 $n-d$，其中 $f(x) = \sum_{i=0}^{n-1} A_i x^i$。 定义 $\Phi_n(x)$ 为 $\prod_{d\|n} \Phi_d(x) = x^n - 1$，那么可以发现 $\Phi_x(x)$ 无法表示为两个次数均不为 0 的多项式的乘积。所以我们只需要判断是否存在一个 n 的约数 d 使得 $\Phi_d(x)\|f(x)$。这个问题等价于 $f(x) \times \prod_{i\|d\text{且 i 是质数}}(x^{d/i} - 1)$ 被 $x^d - 1$ 整除。所以就可以用简单的模拟来判定了。

时空复杂度	时间复杂度 $O(n \log n + n \times f(n))$，$f(n)$ 是 n 的因子个数，空间复杂度 $O(n)$。

题 36

试题编号	Codechef NOV 13
试题名称	Gangsters of Treeland

题目大意	算法讨论
给定一棵 n 个节点的有根树，开始每一个点都有一个不同的权值，接下来进行 m 次操作：1. 把 i 到根路径上的所有节点标记成一种新的权值。2. 询问 i 子树中所有节点到根路径上不同权值个数的平均值。 数据范围：$n, m \le 2 \times 10^5$	可以发现修改操作相当于 LCT 中的 access 操作，我们对每一种权值记录它最上端的节点，那么因为 access 中 splay 次数是 $O(n \log n)$ 的，所以上端节点的变动次数也是 $O(n \log n)$ 的，于是就可以维护每一种权值最上端的节点。 那么一个节点到根路径上不同权值的个数就相当于路径上有多少个标记节点。接上子树和后可以直接用 DFS 序加树状数组来维护。

时空复杂度	时间复杂度 $O(n \log^2 n)$，空间复杂度 $O(n)$。

题 37

试题编号	Codechef OCT 13
试题名称	Fibonacci Number

题目大意	算法讨论
定义 $F_n = n(n \le 1), F_n = F_{n-1} + F_{n-2}(n > 1)$。现在给定 M, C，询问满足 $F_n \mod M = C$ 的最小的 n，如果无解输出 -1。 数据范围：$T \le 100$,$0 \le C < M \le 2 \times 10^9$,$M$ 为质数且 $M \mod 10$ 为完全平方数。	可以先写出斐波那契数列的通项公式，在题目条件中，2 存在逆元，$\sqrt{5}$ 存在等价的整数。把所有数字都用等价的数字替换之后就变成了解方程 $x^n - x^{-n} = a$，可以解出 $x^n = \frac{a \pm \sqrt{a^2+4}}{2}$，如果 $\sqrt{a^2+4}$ 在模意义下存在值，那么就可以用大步小步得到答案，否则则无解。注意要分 n 的奇偶性讨论。求一个数在模意义下的平方根可以使用 Cipolla's algorithm。

时空复杂度	时间复杂度 $O(\sqrt{P})$，空间复杂度 $O(\sqrt{P})$。

题 38

试题编号	Codechef SEPT 13
试题名称	Two Roads

题目大意	算法讨论
平面上给定 n 个点，现在你要选择两条直线，最小化所有节点到这两条直线上最近点的路径的平方和。保证不存在重点和三点共线。 数据范围 $n \le 100$	可以发现这两条直线夹角的两个角平分线把平面分成了四个区域，每一条直线控制了一组处在对角关系的区域（即这块区域中的点的最近点在这条直线上）。可以发现这两条角平分线一定是垂直的。 如果我们可以确定每一条直线控制的区域，那么就可以通过线性回归来得到答案。可以发现对于每一种角平分线的方案，一定存在一条角平分线经过了两个点（确切说是离这两个点以相反的方向无穷接近），另一条角平分线经过一个点（无穷接近）。于是我们可以枚举所有经过两个点的直线，然后按照其他点到这条线上垂足位置从小到大枚举，这样就可以在 $O(n^3 \log n)$ 的时间内求出每一种可能的划分方案中每一组区域的 $\sum x, \sum y, \sum x^2, \sum y^2, \sum xy$ 以及包含的点数。 知道了这些就可以对每一种划分方案拟合一条最优的直线。大致就是根据点到直线距离公式 $\frac{\lvert Ax+By+C\rvert}{\sqrt{A^2+B^2}}$ 以及最优直线一定经过 $(\overline{x}, \overline{y})$，然后用一元二次方程的判别式计算。

时空复杂度	时间复杂度 $O(n^3 \log n)$，空间复杂度 $O(n)$。

题 39

试题编号	Codechef SEPT 13
试题名称	To Queue or not to Queue

题目大意	算法讨论
开始你有一个空串，接下来进行 n 次操作：1. 在末尾插入一个字符。2. 在开头删掉一个字符。3. 询问当前字符串的不同子串个数。 数据范围 $n \le 10^6$	如果没有删除操作，那么就可以直接用 Ukkonen 算法来计算。只要把 Ukkonen 算法稍微拓展一下就能支持删除操作了，即每一次删除一个最长的后缀。只需要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节点以及合并所有度数为 1 的祖先节点。注意要特判活跃节点的位置。

时空复杂度	时间复杂度 $O(n)$，空间复杂度 $O(n)$。

题 40

试题编号	Codechef JULY 13
试题名称	Across the River

题目大意	算法讨论
给定 n 个点和 m 种圆，每种圆都有一个价格。现在你需要给每一个点购买至多一个圆（这个点作为圆心），使得存在一条只经过圆上的点的路径从直线 $y = d$ 到达 $y = 0$。 数据范围 $T = 10, 1 \le n, m \le 250$	每一个点拆 m 个点表示在这个点买哪一种圆，如果两个圆存在公共点那么就在对应的两个点之间连上一条边，跑最短路就是答案了。可以发现如果 (i, j) 到 (a, b) 存在边，那么对于 $c > b$，一定也存在边到 (a, c)，于是只需要求出满足 (i, j) 到 (a, b) 有边的最小的 b，然后在所有 (i, j) 到 $(i+1, j)$ 之间连边表示扩大在这个点上购买的圆的半径，就能把边数优化到 $O(n^2 m)$ 了。

时空复杂度	时间复杂度 $O(n^2 m \log n)$，空间复杂度 $O(n^2 m)$。

题 41

试题编号	Codechef JUNE 13
试题名称	Two k-Convex Polygons

题目大意	算法讨论
给定 n 根长度给定的木棍，问能否从中选出 $2k$ 根木棍使得它们可以拼成两个边数为 k 的凸多边形。（不能有两条相邻边平行） 数据范围 $1 \le n \le 1000, 1 \le k \le 10$，木棍长度不超过 10^9	一组木棍能拼成凸多边形的条件是最长边小于其它所有边的边长之和。如果只需要拼成一个凸多边形，那么最优的方案一定是把木棍按照长度排序后选取一个连续区间内的木棍。 所以可以分情况讨论：1. 两个凸多边形选取的木棍区间不想交，这一部分直接循环判定即可。2. 两个区间相交，这时选取的一定是一个长度为 $2k$ 的区间，其中假定长度最大的木棍属于第二个凸多边形，我们枚举那个多边形剩下的 $k-1$ 条边的所有可能，然后直接判断。

时空复杂度	时间复杂度 $O(n^2 + n\binom{2k-1}{k})$，空间复杂度 $O(n)$。

题 42

试题编号	Codechef JUNE 13
试题名称	Count Special Matrices

题目大意	算法讨论
一个 $n \times n$ 的矩整数阵是好的当且仅当它满足：$1.A_{i,i} = 0$。$2.1 \le A_{i,j} < n-1 (i \ne j)$。3. 对于每一个整数 $0 \le k \le n-2$，矩阵中都有至少有一个位置的值等于 k。$4.A_{i,j} \le \max(A_{i,k}, Ak, j) (1 \le i, j, k \le n)$ 给定 n，求 $n \times n$ 的好的矩阵的个数对 1000000007 取模后的值。 数据范围 $3 \le n \le 10^7, 1 \le T \le 10^5$	可以得到答案等于 $\frac{n! \times (n-1)!}{3 \times 2^{n-1}}(\frac{3n}{2} - 2 - \sum_{i=1}^{n-1} \frac{1}{i})$。于是就可以通过这个公式直接对所有可能的 n 预处理出答案，然后每组询问直接回答。预处理时要尽可能减少乘法的次数来降低常数。

时空复杂度	时间复杂度 $O(n + T)$，空间复杂度 $O(n)$。

题 43

试题编号	Codechef MAY 13
试题名称	Queries on tree again!

题目大意	算法讨论
给定一棵 n 个节点的带边权的基环外向树，两点之间的最短路径定义为经过点数最少的路径。保证环的大小是奇数，即保证任意两点的最短路径唯一。接下来进行 m 次操作：1. 把两点之间最短路径上的所有边的权值取相反数。2. 询问两点之间最短路径上的边权的最大子段和。 数据范围 $1 \le n, m \le 10^5$	对每一棵外向树维护一个树链剖分，然后再最环上所有边维护一棵线段树，于是每一次修改和询问都可以转化成 $O(1)$ 次树链剖分上的操作和线段树上的操作，即 $O(\log n)$ 次线段树上的操作。特判一下各种情况然后使劲码就好了。

时空复杂度	时间复杂度 $O(n \log^2 n)$，空间复杂度 $O(n)$。

题 44

试题编号	Codechef MARCH 13
试题名称	Little Elephant and Colored Coins

题目大意	算法讨论
给定 n 种给定面值 V_i 和颜色 C_i 的硬币，每种硬币都有无穷多个。接下来有 Q 组询问，你要选出一些硬币使得它们的和为 S，需要最大化选出硬币中的颜色种类数，无解输出 -1。 数据范围 $1 \le n \le 30, 1 \le V_i, Q \le 2 \times 10^5, 1 \le S \le 10^{18}$	如果只需要判断是否存在一种选取硬币的方案，那么就可以先删去 V 最小的硬币，假设它的面值是 m，接下来就可以在模 m 的意义下进行 DP，令 dp_i 为这些硬币可以拼成的模 m 为 i 的最小面值。询问就只需要判断 S 和 $dp_{S \bmod m}$ 的大小关系。 如果要最大化颜色种类数，那么就可以对 DP 再加一维，即 $dp_{i,j}$ 表示使用 i 种颜色的硬币可以拼成的模 m 为 j 的最小面值，注意要特判作为基准的硬币所在的颜色。

时空复杂度	时间复杂度 $O(n^2V)$，空间复杂度 $O(nV)$。

题 45

试题编号	Codechef JAN 13
试题名称	A New Door

题目大意	算法讨论
给定一个矩形区域以及 n 个圆，询问这 n 个圆的圆并的周长在矩形范围内的部分的长度。 数据范围 $n \le 1000$	可以对于每一个圆，求出这个圆在矩形内部且不被其他圆覆盖的周长长度，把这些长度累加起来就是答案了。这可以直接用余弦定理计算出来每一个圆与它相交的区间，然后排序去重求得。需要注意精度问题。

时空复杂度	时间复杂度 $O(n^2 \log n)$，空间复杂度 $O(n)$。

题 46

试题编号	Codechef JUNE 15
试题名称	Chefbook

题目大意	算法讨论
给定 m 个限制 a_i, b_i, W_i, L_i, R_i，你要设定 n 个非负整数 x_i 和 n 个非负整数 Y_i，满足对于没一个限制都有 $L_i \le X_{a_i} - Y_{b_i} + W_i \le R_i$。现在你需要最小化 $\sum_{i=1}^{m} X_{a_i} - Y_{b_i} + W_i$，并输出一种可行的方案。 数据范围 $n \le 100$,$m \le n^2$	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。如果变量 X_i 在限制中出现了 k 遍，就从源到它连一条流量为 k 费用为 0 的边，对于变量 Y_i 就连向汇。这张图的最小费用最大流就是可行的答案了。 这样通过最小费用最大流得到的答案是从原问题的对偶问题得到的。考虑用差分约束系统构造一个解，如果只把初始限制连入那么差分约束系统可以得到一个可行解，考虑增加限制，对于图中的每一条代表限制的边，如果它的流量不为 0，说明这一个限制一定被刚好达到，于是对于所有流量非 0 的边都在查分约束系统中加入这个限制后即可得到一组合法的最优解。

时空复杂度	时间复杂度 $O(maxflow(n,m) + nm^2)$，空间复杂度 $O(m)$。

题 47

试题编号	Codechef AUG 15
试题名称	Future of draughts

题目大意	算法讨论
给定 T 张 n_i 个点 m_i 条边的无向图，有 Q 组询问，每组询问问只考虑编号为 $[L_i, R_i]$ 的图，最开始对每一张图可以选择一个出发点，接下来每一个回合可以选中一些图（至少选中一个），并对每一个选中的图通过一条存在的边移动一个位置，问在 K_i 回合内每一张图都回到出发点的方案数（对 1000000007 取模）。 数据范围 $n,T \le 50,K_i \le 10^4,Q \le 10^5$	可以先考虑预处理每一张图长度为 $i(1 \le i \le maxK)$ 的回路个数。显然长度为 i 的回路个数等于邻接矩阵 G 的 i 次方的对角线上元素的和。可以先对这个邻接矩阵求出它的特征多项式，这个可以根据特征多项式的定义 $\det(xI-G)$，对 x 带入 n 个不同的值，分别求出行列式，然后再插值来得到特征多项式 $f(x)$。因为 $f(G)=0$，所以就可以得到一个关于回路个数的 n 阶线性递推式，这样就可以预处理出长度为 i 的回路个数了。 对于每一组询问，令 $w_{i,j}$ 为第 i 张图长度为 j 的回路个数，通过简单的容斥可以得到恰好在第 K 回合结束的方案数是 $\sum_{i=1}^{K}\prod_{j=L}^{R} w_{j,K} \times \binom{K}{i} \times (-1)^{K-i}$，答案就是这个的前缀和。于是可以枚举 L 和 R，然后用 FFT 预处理出答案。因为模数是 10^9+7，所以需要选取三个 NTT 模数分别求值然后合并得到答案。

时空复杂度	时间复杂度 $O(Tn^4 + TnK + T^2K\log K)$，空间复杂度 $O(Tm+K)$。

题 48

试题编号	Codechef JAN 13
试题名称	Cucumber Boy and Cucumber Girl

题目大意	算法讨论
给定 B 个 $n \times n$ 的矩阵 A_i，对于数对 $(a,b)(a<b)$ 定义 $n \times n$ 的矩阵 B 满足 $B_{i,j} = \sum_{k=1}^{n} A_{a,i,k} \times A_{b,j,k}$，一个 $1-n$ 的排列 P 是好的当且仅当至少存在一个 i 使得 B_{i,p_i} 是奇数，数对 (a,b) 是好的当且仅当好的排列有奇数个。 询问有多少个好数对。 数据范围 $n \le 60, B \le 8000$	对于矩阵 B，令矩阵 C 为 $C_{i,j} = (B_{i,j}+1) \mod 2$。考虑矩阵 C 的行列式，每一个不是好的排列的贡献都是 1 或 -1，每一个好的排列的贡献都是 0，当 $n>1$ 的时候数对 (a,b) 是好的等价于 $\det(C)$ 为奇数。因为 $C_{i,j} = (\sum_{k=1}^{n} A_{a,i,k} \times A_{b,j,k} + 1) \mod 2$，所以可以在每一个矩阵 A 后面补上全是 1 的第 $n+1$ 列，那么 $C = A_a \times A_b^T$（模 2 意义下）。 考虑 $\det(C)$，由 Binet-Cauchy 定理，令 $A_{i,j}$ 为矩阵 A 删掉第 j 列后得到的矩阵，那么就有 $\det(C) = \sum_{i=1}^{n+1} \det(A_{a,i}) + \det(A_{b,i})$。所以只需要求出所有的 $\det(A_{a,i})$ 即可。我们可以对 A_i 进行消元，如果 A_i 不满秩，那么 $\det(A_{i,j})$ 一定为 0。否则一定可以把 A 消成 $n \times n$ 的单位矩阵加上一列的形式（把消元后矩阵定义为 D），假设加上的是第 k 列，那么就有 $\det(A_{i,j}) = D_{j,k}(j<k), \det(A_{i,k}) = 1, \det(A_{i,j}) = 0(j>k)$。于是就可以直接统计答案了，上述做法的消元部分和统计部分都可以用 bitset 优化。

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

题 49

试题编号	Codechef DEC 12
试题名称	Different Trips

题目大意	算法讨论
给定一棵 n 个节点的树，每一个节点的权值定义为它的度数，两条路径被视为相同的当且仅当它们长度相同且第 i 个经过的点的权值相同。询问这棵树有多少条不同的从孩子走向祖先的路径 数据范围 $n \leq 10^5$	可以仿照倍增求 SA 的方法（当然这题用 SAM 直接就可以做但是我不会证复杂度），对于每一个节点，拿出它向上走 2^i 个节点的这一段字符串，并按照它们的大小关系进行重新标号。大小关系可以直接通过 2^{i-1} 的标号得到。这样当 $2^i \geq n$ 的时候就把所有节点到根的路径排好了序，那么答案就是所有节点的深度之和减去排好序后相邻两个节点根路径的 LCP，这个可以直接倍增得到。

时空复杂度	时间复杂度 $O(n \log n)$，空间复杂度 $O(n \log n)$。

题 50

试题编号	Codechef NOV 14
试题名称	Sereja and Order

题目大意	算法讨论
有 n 个程序和两台电脑，第 i 个程序在第一台电脑上要运行 A_i 秒，在第二台电脑上运行 B_i 秒，现在要在最少的时间内完成所有程序在两台电脑上的运行任务，求出最少时间并输出一个方案。 数据范围 $n \leq 10^5$	答案的下界是 $\max(\sum A_i, \sum B_i, A_i + B_i)$，可以发现这个下界是肯定可以取到的。如果下界是取在 $A_i + B_i$，那么直接把剩下的值填进去就好了。否则一定有一台计算机是不间断地在运行，可以发现最优解的取值一定非常多，所以可以尝试随机，如果我们已经得到了两台计算机运行程序的顺序，那么就可以通过简单的贪心得到运行完的最早时间，如果运行时间等于答案的下界，说明我们已经得到了一个解。

时空复杂度	时间复杂度 $O(n \times \text{玄学})$，空间复杂度 $O(n)$。

题 51

试题编号	Codechef OCT 11
试题名称	The Baking Business

题目大意	算法讨论
有 n 个操作，每一个操作都可能是增加订单或者询问，每一个订单用如下的方式描述： 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 顾客的性别为 M 或者 F，年龄从 1 到 90。注意所有的编号都是从 0 开始。还需要注意方括号内的部分是可选的，因为某些出售的这些信息丢失了。出售数量不会超过 100。 每一个询问都可以用如下的方式描述： 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] 询问该限制下的出售总数。如果方括号内的信息缺失说明对这个信息没有限制，特殊地如果产品编号省编号是 -1 说明没有限制。 数据范围：有 10 种产品，每种都有 3 种不同的大小。有 10 个省份，每个省份可以被划分为 20 个城市，每个城市又可以被划分成 5 个地区。 $n \leq 100000$	可以开一个七维数组来记录所有的订单，其中每一维为 0 表示没有限制，然后对于每一个插入的订单在这个数组中更新。为了要支持查询，需要对年龄一维记录前缀和。

时空复杂度	时间复杂度 $O(1)$，空间复杂度 $O(n)$，常数巨大。

题 52

试题编号	Codechef OCT 11
试题名称	Sine Partition Function Solved

题目大意	算法讨论
给定 n, m, K，问 $\sum_k \prod_{i=1}^{m} \sin(k_i K)$，其中 k_i 为非负整数 数据范围：$m \le 50, n \le 10^9, 0 \le K \le 6.28$，保证答案不超过 10^{300}，精度需要达到绝对或相对误差不超过 0.1。	考虑最简单的 DP，定义： $f_{n,m} = \sum_k \prod_{i=1}^{m} \sin(k_i K)$ $g_{n,m} = \sum_k \cos(k_m K) \prod_{i=1}^{m-1} \sin(k_i K)$ 那么根据合角公式，就可以得到递推式： $f_{n,m} = (f_{n-1,m-1} + g_{n-1,m}) \sin(K) + f_{n-1,m} \cos(K)$ $f_{n,m} = (f_{n-1,m-1} + g_{n-1,m}) \cos(K) - f_{n-1,m} \sin(K)$ 于是就可以使用矩阵乘法来优化递推得到答案。

时空复杂度	时间复杂度 $O(m^3 \log n)$，空间复杂度 $O(m^3)$。

题 53

试题编号	Codechef AUG 15
试题名称	Simple Queries

题目大意	算法讨论
给定一个长度为 n 的数列和 m 个操作： 1. 定义 S 为区间 $[L, R]$ 中出现过的数字的集合，求 $\sum_{1 \le i < j < k \le \|S\|} S_i S_j S_k \bmod 1000000007$。2. 插入一个数。3. 删除一个数。4. 修改一个位置的值。5. 询问一个区间内出现过的数字种类数 数据范围：$n, m \le 10^5$	把每一个位置标记为 (x, y)，其中 x 为它的下标，y 为它后面第一个数字和它相同的位置的下标，如果没有就是数列长度 +1。事先用平衡树预处理，给每一个插入操作分配一个不变的下标，这样就可以避免下标的变化，可以直接用树状数组维护了。可以用树状数组套线段树来维护这些点，对于询问操作只需要询问 $x \in [L, R], y > R$ 的点的信息。询问 5 是询问点数，询问 1 只需要求出这些点的权值和 A, 平方和 B，三次方和 C，那么答案就是 $\frac{A^3 - 3AB + 2C}{6}$。至于修改，可以对每一个数字开一个 set 来维护它出现的下标，然后当出现变化的时候在 set 上修改（查询）并更新到树套树上即可。

时空复杂度	时间复杂度 $O(m \log^2 n)$，空间复杂度 $O(m \log^2 n)$。

题 54

试题编号	Codechef SEP 12
试题名称	Knight Moving

题目大意	算法讨论
一个无限延生的棋盘，你开始有一个骑士在 $(0, 0)$。如果骑士的坐标为 (x, y)，那么下一个回合你可以选择把骑士移动到 $(x + A_x, y + A_y)$ 或者 $(x + B_x, y + B_y)$。同时棋盘上有 K 个位置是障碍物，骑士不能移动到那些位置上。问有多少种方案把骑士移动到坐标 (X, Y)，对 1000000007 取模，如果有无穷多种方案输出 -1。 数据范围：$K \le 15$，其余数字的绝对值不超过 $d = 500$。	如果向量 (A_x, A_y) 和 (B_x, B_y) 线性无关，那么每一个格子都可以唯一地表示成一个数对 (a, b)，表示走 a 步 (A_x, A_y) 和 b 步 (B_x, B_y) 后能被到达，如果不考虑障碍物，那么方案数就是 $\binom{a+b}{a}$。既然有障碍物就只需要容斥一下就好了。 如果向量 (A_x, A_y) 和 (B_x, B_y) 线性相关，那么问题就转化为了一维的问题，可以发现和问题有关的坐标范围只有正负 500^2，所以只要把每一个点映射到数轴上然后暴力 DP 就好了。

时空复杂度	时间复杂度 $O(K^2 2^K + d^2)$，空间复杂度 $O(d^2)$。

题 55

试题编号	Codechef AUG 12
试题名称	Two Magicians

题目大意	算法讨论
给定一张 n 个点 m 条边的简单无向图，两个人博弈，最开始两个人分别在 1 号点和 2 号点。从第一个人开始轮流操作，每一个回合有以下三个步骤：1. 可以沿着现有的无向边移动任意步，如果这一步结束时两个人在同一个格子，则当前人胜。2. 加入一条连接两个节点现在还没有的无向边，如果无法加入则另一个人胜。3. 最开始每一个人有 P 次传送机会，如果当前人还有传送的机会，他可以选择消耗一次并传送到任意一个节点。问谁必胜。 数据范围：$T \leq 10^2, n \leq 7777, m, p \leq 10^5$	考虑 DP 求解，可以发现当前博弈的状态可以简化为：两人所在联通块大小的奇偶性，两人剩下的传送次数，奇数块的个数，偶数块的个数以及添加后不会影响连通性的边的条数。只需要枚举每一次操作后可能的状态转移即可，但是这样时间复杂度是 $O(n^2P^2)$ 的。 可以发现如下的规律：任何的游戏中每一个人至多使用一次传送，当奇数块足够多 (10 以上) 的时候 DP 值以 4 为周期循环，当偶数块足够多（10 以上）的时候 DP 值不变。所以状态数就能被简化为 $O(1)$ 了。 因此只需要最开始先跑一遍 DP，然后对每一组数据输出联通块个数带入找到值即可。

时空复杂度	时间复杂度 $O(n\alpha n)$，空间复杂度 $O(n+m)$。

题 56

<table>
<tr><td>试题编号</td><td colspan="2">Codechef AUG 12</td></tr>
<tr><td>试题名称</td><td colspan="2">A Game of Thrones</td></tr>
<tr><td colspan="2">题目大意</td><td>算法讨论</td></tr>
<tr><td colspan="2">给定一个序列，序列中有 n 种数，第 i 种的权值是 u_i，出现了 c_i 次。两个人博弈轮流操作。第一个回合第一个选取一个数字作为初始的局面值。之后的每一个回合，当前的人需要选出一个和局面值相似的数作为局面的值，之前的局面值讲移出游戏（如果有多个只移出一个），如果无法操作则算输。两个数 $a, b(a > b)$ 是相似的当且仅当 $b|a$ 且 $\frac{a}{b}$ 是质数。
你需要判断哪个人必胜，如果是第一个人必胜，你需要输出可以使他获胜的最大的初始局面值是多少。
数据范围：$n \le 500, u_i \le 10^{18}, c_i \le 10^9$</td><td>如果两个数是相似的，我们在它们之间连上一条边（这一步可以使用 miller rabin 来实现），那么第二个人必胜当且仅当这张图存在完备匹配。因为这一张图一定是二分图，所以只需要把图建出来然后跑一遍最大流就能知道是谁获胜了。
可以发现，如果一个初始权值可以使第一个获胜，那么必定存在一个最大匹配使得这个数没有匹配边。因为对每一个值跑一遍最大流太慢了，所以可以用退流来判断。</td></tr>
<tr><td>时空复杂度</td><td colspan="2">时间复杂度 $O(n^2 \log n + maxflow(n, n \log n))$，空间复杂度 $O(n \log n)$。</td></tr>
</table>

题 57

<table>
<tr><td>试题编号</td><td colspan="2">Codechef JULY 12</td></tr>
<tr><td>试题名称</td><td colspan="2">Dynamic GCD</td></tr>
<tr><td colspan="2">题目大意</td><td>算法讨论</td></tr>
<tr><td colspan="2">给定一棵 n 个节点的带点权的树，有 m 次操作：1. 询问两点之间路径上所有节点权值的最大公约数。2. 给两点之间路径上所有点的权值加上一个值。
数据范围：$n, m \le 5 \times 10^4$</td><td>首先对序列进行树链剖分然后求 DFS 序，这样就转化为了序列问题。可以发现有 $\gcd(a, b, c) = \gcd(a, b - a, c - b)$，所以可以对序列差分，分别维护每一个节点的值以及差分后的区间 gcd，这可以用线段树非常容易地实现。</td></tr>
<tr><td>时空复杂度</td><td colspan="2">时间复杂度 $O(m \log^3 n)$，空间复杂度 $O(n)$。</td></tr>
</table>

题 58

<table>
<tr><td>试题编号</td><td colspan="2">Codechef MAY 12</td></tr>
<tr><td>试题名称</td><td colspan="2">Little Elephant and Boxes</td></tr>
<tr><td colspan="2">题目大意</td><td>算法讨论</td></tr>
<tr><td colspan="2">有 n 个盒子，第 i 个盒子里有 P_i 的概率是 V_i 块钱，$1 - P_i$ 的概率是一块钻石。你打开了所有盒子之后去买东西，一共有 m 件物品，第 i 件需要 C_i 块钱和 D_i 个钻石，你一定会买数量尽可能多的物品。问你期望能买到多少件物品。
数据范围：$n, m, D_i \le 30, V_i, C_i \le 10^7$</td><td>可以使用 meet in middle 的策略，先枚举前 x 个盒子的所有情况，记录下来开前 x 个盒子得到 a 块钱 b 块钻石的概率并根据 b 分别记录到 x 个数组中，再对 a 排序处理出前缀和。然后可以用背包求出用 a 块钻石买 b 个物品最少需要多少钱。
在这之后再搜后 $n - x$ 个盒子的情况，对于每一种情况我们枚举一共有多少个钻石以及买了多少个物品，这样就可以在预处理的数组中进行二分求得这种情况下得概率。当 x 取 20 左右得时候就能通过所有测试点了。</td></tr>
<tr><td>时空复杂度</td><td colspan="2">时间复杂度 $O(x2^x + n^3 + n^2 x 2^{n-x})$，空间复杂度 $O(2^x + nm)$。</td></tr>
</table>

题 59

试题编号	Codechef MAY 12	
试题名称	Selling Tickets	
题目大意		算法讨论
有 n 道菜 m 个人，这些人中有一部分要来参加晚餐，晚餐时每一道菜都被分配给一个人，第 i 个人只要吃到第 a_i 道菜或第 b_i 道菜就能开心，否则他就不开心。现在求最大的 x 使得任意的 x 人来才加晚餐都存在一个分配的方案使所有人都开心。 数据范围：$n \le 200, m \le 500$		问题可以转化为给定一张 n 个点 m 条边的无向图，问边数恰好比点数多一的联通子图的点数最少是多少。可以发现这样的子图只有两种情况： 1. 两点之间的三条路径，这种情况可以直接枚举这两个点然后用简单的 BFS 来求解。 2. 两个由一条路径连接起来的简单环，此时可以枚举其中一个度数至少为 3 的点并以它为根求 BFS 树，此时每一个简单环代表了树上的一条路径且每一个点到树根的最短路径都是树边，我们把每一个环的权值设为它的边数加上它两端的 LCA 到根的距离，只需要拿此时权值最小的两个环的权值之和更新答案就行了。
时空复杂度	时间复杂度 $O(n^2m)$，空间复杂度 $O(n+m)$。	

题 60

试题编号	Codechef JAN 12	
试题名称	Card Shuffle	
题目大意		算法讨论
n 张卡片，最开始从 1 到 n 从上到下摆放着，接下来有 m 次操作，每一次取出前 A 张，再取出前 B 张，然后把前 A 张放回去再取出前 C 张，接着把 B 倒序放回去，最后把 C 放回。问最后的卡牌顺序数据范围：$n, m \le 10^5$		直接用最基本的平衡树打一个区间翻转标记来模拟就行了。
时空复杂度	时间复杂度 $O(m \log n)$，空间复杂度 $O(n)$。	

题 61

试题编号	Codechef APRIL 12	
试题名称	Substrings on a Tree	
题目大意		算法讨论
给定一棵 n 个节点的树，每一个节点都被标记了一个小写字母。一个字符串被这棵树表示了当且仅当它可以被表示为一个点往它的后代移动路径上的经过所有点的字母连接起来得到的字符串。求有多少个字符串被这颗树表示了。 之后有 m 次询问，每一次询问给出了 26 个字母的大小顺序，求被这棵树表示的字符串中第 K_i 小的字符串。 数据范围：$n \le 2.5 \times 10^5, m \le 5 \times 10^4$		对这一棵树建出后缀自动机，只要边 BFS 边插入即可。因为后缀自动机是一个 DAG，每一个能被表示的字符串都可以被看成从出发点到某一个节点的一条路径，所以第一问统计的就是路径数。这只需要对后缀自动机拓扑排序然后递推一遍就好了。至于第二问，我们对每一个节点预处理出从它出发有多少条不同的路径，之后就可以 DFS 得到答案了。
时空复杂度	时间复杂度 $O(m \log n)$，空间复杂度 $O(n)$。	

题 62

试题编号	Codechef APRIL 12
试题名称	Find a special connected block
题目大意	算法讨论
一个 $n \times m$ 的网格，每一个格子都有一个 $[-1, n \times m]$ 范围内的整数权值以及一个代价。问一个权值和最小的四联通块满足联通块中没有权值为 -1 的格子且至少出现了 k 种不同的正权值。 数据范围：$n, m \le 15, k \le 7$	如果每一个格子的权值范围在 $[-1, k]$ 范围内，那么它就是一个经典的斯坦纳树问题，令 $dp_{i,j}$ 为以第 i 个点为根包含的颜色集合为 j 的最小代价和，分别用枚举子集和最短路转移就行了。 对于 $n \times m$ 种权值，我们随机一种 $[1, n \times m]$ 到 $[1, k]$ 的映射，并对映射结束后的表格求解最优值，那么得到的答案一定不会比原问题的最优解优且得到原问题的解的概率是 $\frac{k!}{k^k}$，于是我们可以随机 x 次取最优解。
时空复杂度	时间复杂度 $O(xnm3^k + xnm\log n2^k)$，空间复杂度 $O(nm)$。

题 63

试题编号	Codechef MARCH 12
试题名称	Evil Book
题目大意	算法讨论
有 n 个人，你打败第 i 个人需要付出 c_i 的代价，打败它后可以获得 d_i 的魔法值，最开始你的魔法值是 0。你可以对人使用魔法，对第 i 个人使用后 c_i 和 d_i 都将除以 3（实数），魔法可以使用无限多次但是每一次使用要消耗 X 点魔法值，魔法值不够的之后不能使用魔法。 你要使你的魔法值大于等于 666，问你最少付出多少的代价（打败每一个人代价累加）。 数据范围：$T \le 5, n \le 10, 10 \le X \le 666$	除了第一个人以外，之后你战胜每一个人一定会把它的魔法值除到 666 以下，如果你用了 i 次魔法，那么一定有 $Xi < \frac{d}{3^i}$，可以得到你对每一个使用魔法的数目范围在一个长度不超过 4 的区间内，所以我们可以使用搜索。 可以发现一定存在一种最优的方案使得对每一个人使用的魔法次数单调不减，于是搜索的状态就是：当前至少使用多少次，哪些人还没有被战胜，剩余的魔法值，已经付出的代价。加一些剪枝就好了。
时空复杂度	时间复杂度 $O(T4^n)$，空间复杂度 $O(n)$。

题 64

试题编号	Codechef MARCH 12
试题名称	Ciel and Earthquake
题目大意	算法讨论
一个 $n \times m$ 的网格，两个格子之间存在一条边当且仅当它们有一条公共边。现在每一条边都有 p 的概率损毁，问点 $(1,1)$ 和点 (n,m) 联通的概率是多少。 数据范围：$T \le 50, n \le 8, m \le 10^{18}$	当 m 不大的时候，可以使用轮廓线 DP 来求解，状压轮廓线上的每一个点和 $(1,1)$ 的连通性，经过实践联通性的数目 S 只有 3000 多个，预处理转移之后可以使用 $O(nmS)$ 的 DP 来求解一组的答案。 令 $S(n,m,p)$ 等于输入为 n,m,p 时的答案，当 m 很大时，我们可以选取一个足够大的 x，这时存在关系 $(\frac{S(n,x+1,p)}{S(n,x,p)})^{m-x} \times S(n,x,p) \approx S(n,m,p)$，当 x 取 40 左右的时候就能满足题目的精度要求了。
时空复杂度	时间复杂度 $O(TnxS)$，空间复杂度 $O(nxS)$。

题 65

试题编号	Codechef DEC 11
试题名称	Short II

题目大意	算法讨论
给定质数 p，问有多少对 $a, b(a > p, b > p)$ 满足 ab 被 $(a-p)(b-p)$ 整除。 数据范围：$T \le 5, p \le 10^{12}$	可以把原问题等价于求满足 $ab \mid p(a+b+p)$ 的数对个数而不改变答案，可以分三种情况讨论： 1.ab 同时被 p 整除，这个时候满足条件的 a 和 b 只有 5 对。 2.ab 同时不被 p 整除，这时显然有 $a \ne b$，我们假定 $a < b$，这时候满足 $a < 1 + \sqrt{p+1}$，此时 $b = \frac{a+p}{ak-1}$，其中 k 是满足条件的任意整数，令 $d = ak - 1$，那么一个合法解需要满足以下条件：1.$b > a$ 2.a 不被 p 整除 3.$d \mid (a+p)$ 4.$a \mid d+1$。我们分 $b \le d$ 和 $d \le b$ 两种情况枚举较小的数然后直接检查这些条件，可以发现这两个枚举的上届都是 $\sqrt{p+1+\sqrt{p+1}}$，这样就可以求得这一部分的解了。 3.ab 中恰好有一个被 p 整除，可以发现这种条件的对数是 2 的两倍，因为对于任意一个满足 2 条件的数对 (a, b)，都可以得到恰好两个满足 3 条件的数对 $(a, \frac{p(a+p)}{b})$ 和 $\frac{p(b+p)}{a}, b$。

时空复杂度	时间复杂度 $O(T\sqrt{p})$，空间复杂度 $O(1)$。

题 66

试题编号	Codechef NOV 11
试题名称	Luckdays

题目大意	算法讨论
给定整数 A, B, X, Y, Z, P, C，按照以下的方式生成序列 S：$S_1 = A$,$S_2 = B$,$S_i = (XS_{i-1} + YS_{i-2} + Z) \mod P$。接下来有 Q 组询问，每组询问给出 L_i, R_i，你需要求出满足 $i \in [L_i, R_i]$ 且 $S_i = C$ 的整数 i 的个数。 数据范围：$T \le 2, Q \le 2 \times 10^4, p \le 10007$ 且 p 是个质数，$1 \le L_i, R_i \le 10^{18}$。	当 X, Y 中有一个模 p 为 0 的时候，序列 S 的循环节长度不会超过 p，所以可以直接暴力求解。接下来讨论 X 和 Y 均不为 0 的情况。 此时序列的循环节长度可以达到 p^2。如果是要求 S 的某一项，可以直接使用矩阵乘法来求解，矩阵乘法的结果是一个 1×3 的矩阵 $(S_i, S_{i+1}, 1)$。现在我们要解方程 $S_i = C$，即相当于求满足矩阵为 $(C, x, 1)$ 的 i 的个数，其中 $x \in [0, p)$。 此时我们可以用大步小步来求解，对 0 到 p^2 中的所有数，每 $\sqrt{p}$ 个求出它所代表的矩阵存到哈希表中，然后枚举 x，对于每一个可能的矩阵，我们给它依次乘以 $\sqrt{p}$ 次矩阵，并在 hash 表中查找。这样就可以求出第一个循环节内所有满足 $S_i = C$ 的 i。 同理我们可以用这个方法求出序列的循环节。有了这两个信息就可以非常简单的得到答案了。

时空复杂度	时间复杂度 $O(Tp\sqrt{p} + TQ \log p)$，空间复杂度 $O(p\sqrt{p})$。

题 67

试题编号	Codechef FEB 12
试题名称	Find a Subsequence

题目大意	算法讨论
给定一个长度为 5 的排列 p 和一个长度为 n 的整数序列 A，现在要求一个 A 的长度为 5 的子序列 B 满足在序列 B 中恰好有 $p_i - 1$ 个数比 B_i 小。 数据范围：$T \le 60, n \le 2000, A_i \le [-10^9, 10^9]$	枚举第二个数和第四个数下标，此时我们可以贪心地选取第一个数的值和第五个数的值。即如果它比第三个数大，那么它就取所有合法的选取方案中值最大的那个数，如果它比第三个数小，那么它就取所有合法的选取方案中值最小的那个数。得到这四个数之后就相当于得到了一个对第三个数的取值范围和下标范围的约束，如果在这个范围内存在可能的数，我们就找到了一个合法的子序列。至于子序列的下标我们可以通过值暴力一遍得到。 上述做法用到的信息有前缀（后缀）中比给定值大（小）的最小（大）的数以及一个区间范围内值在一个范围内的数的个数，这个都可以通过 $O(n^2)$ 的预处理得到。

时空复杂度	时间复杂度 $O(Tn^2)$，空间复杂度 $O(n^2)$。

题 68

试题编号	Codechef SEP 11
试题名称	Short

题目大意	算法讨论
给定 n, k，求满足 $n < a, b < k$ 的数对 (a, b) 使得 $(a-n)(b-n) \mid ab-n$。 数据范围：$0 \le n \le 10^5, n < k \le 10^{18}$	不妨假设 $a \ge b$，那么可以得到 $b = n + \frac{n(a-n)}{p(a-n)-a}$，可以枚举所有可能的 a，再枚举 $n(a-n)$ 的所有约数 d，这样就可以求得 $p = \frac{d+a}{a-n}$ 和 b，判断是否合法之后就可以累加进答案了。 可以发现 $d + a \ge 2(a-n)$，所以就有 $b \le n + \frac{n(a-1)}{a-2n}$，又有 $a \le b$，所以就能得到 $a \le 2n + \sqrt{2n^2 - n}$。于是 a 就只需要枚举 $O(n)$ 个值即可。 至于枚举 d，我们可以预处理出所有可能的 $a-n$ 的质因数分解，然后对于每一个 $n(a-n)$ 求出它的所有约数进行判断。但是这样还是会超时。 可以发现当 a 比较大的时候，因为从 $a \le b$ 可以得到 $p \le \frac{a^2-n}{(a-n)^2}$，所以可以直接枚举所有可能的 p，然后解出 d 来判断是否合法。

时空复杂度	时间复杂度 $O(Tnlim)$，其中 lim 为两个求解方法切换时枚举量的阈值，当在 $a = n + 3000$ 的时候切换时 $lim = 1177$，空间复杂度 $O(n \log n)$。

题 69

试题编号	Codechef SEP 11
试题名称	Counting Hexagons

题目大意	算法讨论
现在你有 NK 根木棍，木棍长度为 1 到 N 且每种长度的木棍有 K 根。你需要选出六根木棍拼出一个面积为正的六边形。你选取的木棍需要满足：最长的木棍长度至少为 L，其它的木棍长度不能超过 X。求方案数对 1000000007 取模，两个方案是不同的当且仅当存在一个长度的木棍在两个方案中的选取个数不同。 数据范围：$1 \le L \le 5, 1 \le X < L \le N \le 10^9$ 且 $N - L \le 100$	因为 $N - L$ 很小，所以可以直接枚举最长的木棍的长度。拼出的六边形的面积为正当且仅当其他木棍的长度之和大于最长的木棍的长度。 我们可以通过一个数位 dp 来得到答案，令 $dp[a][b][c][d][e]$ 表示当前考虑了后 a 位的取值，就后 a 位来说其他木棍的长度之和和最长木棍的长度的大小关系为 b，其他木棍的长度之和对第 $a+1$ 位的进位为 c，这五根木棍的大小情况为 d（因为相同的不能超过 K 根），这些木棍就后 a 位来说和 X 的大小关系为 e，转移就只需要枚举这 5 根木棍第 $a+1$ 位是 0 还是 1 即可。 为了避免长度等于 0 的木棍出现，可以先把 X 减去 1，最长的木棍长度减去 1，即在最后给每一个其他的木棍长度增加 1。

时空复杂度	时间复杂度 $O(a(N-L)\log Na2^2Bell_a)$，空间复杂度 $O(a(N-L)2^aBell_a)$，其中 $a = 5$，因为有一些无用状态所以常数还是很小的

题 70

试题编号	Codechef JUNE 11
试题名称	Attack of the Clones

题目大意	算法讨论
我们称一个形为 $f : A \to B$ 的函数叫做布尔函数，其中 A 是一个长度 n 的仅由 0 和 1 组成的数列的集合，$B = \{0, 1\}$，我们称 n 为布尔函数的项数。现在有四个是 n 项布尔函数的集合： 1.Z 集合是所有满徐 $f(0,0,...,0) = 0$ 的集合。 2.P 集合是所有满足 $f(1,1,1,...,1) = 1$ 的集合。 3.D 是满足 $!f(x_1, x_2, ..., x_n) = f(!x_1, !x_2, ..., !x_n)$ 的集合 4.A 集合是满足如下条件的集合： 如果 $f(x_1, ..., x_{i-1}, a, x_{i+1}, .., x_n) = f(x_1, ..., x_{i-1}, b, x_{i+1}, .., x_n)$ 那么 $f(y_1, ..., y_{i-1}, a, y_{i+1}, .., y_n) = f(y_1, ..., y_{i-1}, b, y_{i+1}, .., y_n)$ 对任意 y 成立。 现在给出一个由 $Z, P, D, A, v, \hat{}, !, (,), /$ 组成的合法表达式，其中 $v, \hat{}, !, /$ 分别表示并集，交集，对全集的补集和差集。其中 () 优先级最高，其次是 !，剩下三个优先级相同。这个表达式的结果是一个以函数为元素的集合，求出这个集合的元素个数。 数据范围：$T \le 100$，表达式 S 最多 100 个字符，$n \le 100$，所有询问的 n 相同。	对于任意一个函数，我们可以用一个长度为 4 的二进制数来表示他，每一位代表这个函数是不是属于这一位所代表的集合。我们最开始可以先求出每一个二进制数所代表的集合个数。 接着，我们可以用一个 16 位的二进制数来表示这 16 个部分是不是属于一个集合。那么就可以把集合的运算变成实数的运算。v 变成了 or，ˆ变成了 and，$!A$ 变成了 $65536-a$，/ 变成了 $a-(ab)$。用栈处理这个表达式，最后把满足表达式的部分的贡献累加起来就得到答案了。

时空复杂度	时间复杂度 $O(T\|S\|)$，空间复杂度 $O(\|S\|)$

题 71

试题编号	Codechef AUG 11
试题名称	Shortest Circuit Evaluation
题目大意	算法讨论
布尔表达式满足短路运算原理，即如果某个表达式已经得到正确的结果，那么就不再继续计算这个表达式。 现在给出这个表达式以及每一个变量（只出现一次），你可以在表达式本质不变的情况下调整这个布尔表达式的顺序，使得期望的计算次数最少，求期望概率。 数据范围：表达式 S 长度不超过 30000，变量个数不超过 1000，变量名长度不超过 5，一个变量为 1 的概率 p 满足 $0 < p < 1$	因为 的优先级大于 or，所以我们可以对这个表达式建出一棵表达式树，其中同一个节点的所有孩子代表这由用一种运算符（ 或者 or）依次连接的表达式，可以发现这题中所谓的化简只可能是调整同一个节点的所有孩子的顺序。 对每一个节点我们维护两个值 w_i 和 f_i，w_i 为计算这个子树的期望运算次数，f_i 为这个子树的运算结果为 1 的期望概率。那么当 i 的所有孩子 j 是由 or 连接的时候，按照 $\frac{w_i}{f_i}$ 递增的顺序运算最优，由 连接的时候按照 $\frac{w_i}{1-f_i}$ 递增的顺序运算最优。只需要贪心地 DFS 一遍就好了。
时空复杂度	时间复杂度 $O(\|S\| \log \|S\|)$，空间复杂度 $O(\|S\|)$

题 72

试题编号	Codechef OCT 12
试题名称	Max Circumference
题目大意	算法讨论
给定二维平面中的三个点 ABC 和 N 个操作，第 i 个操作有两个参数 x_i 和 y_i，使用这个操作可以使得点 A 的 x 坐标增加 x_i，并且 y 坐标增加 y_i。你可以使用最多 K 个操作，每一个操作最多使用一次，ABC 三个点允许共线或者重合。 现在你需要最大化 $\|AB\|+\|BC\|+\|AC\|$，答案的绝对误差必须小于 10^{-12} 数据范围：$K \le N \le 500$，坐标范围 10^9，$\|x_i\|, \|y_i\| \le 10^6$	因为 B 和 C 的坐标不变，所以最后要最大化的是 $\|AB\| + \|AC\|$，可以证明一定存在两个数 u, v 使得在最大化 $uA_x + vA_y$ 的同时，$\|AB\| + \|AC\|$ 也最大化。证明很简单，假设最大值为 t，那么可以得到一个椭圆 $\|AB\| + \|AC\| = t$，不存在一个合法的操作使得 A 到达这个椭圆之外，我们过最优的 A 作这个椭圆的切线，这条线的斜率就是满足要求的 u 和 v。 如果已经知道了 u 和 v，那么排序之后取前 K 个并去掉贡献为负的操作算出 A 的坐标，就能得到答案了。我们把所有满足 $ux_i + vy_i = 0$ 或者 $ux_i + vy_i = ux_j + vy_j$ 的 (u, v) 称作关键点，关键点把所有 (u, v) 分成了 $O(n^2)$ 个部分，每一个部分的顺序都是一样的。所以只需要从某一个位置出发顺时针枚举每一个部分然后更新操作的贡献顺序，同时维护一下 x_i 和 y_i 的前缀和，每一次二分得到前 K 个中最后一个操贡献非负的操作就好了。 因为要求绝对误差不超过 10^{-12}，所以直接开方精度无法达到要求，我们可以令 $\sqrt{S} = I + D$，那么使用公式 $D = \frac{S-I^2}{I+\sqrt{S}}$ 来得到答案。
时空复杂度	时间复杂度 $O(N^2 \log N)$，空间复杂度 $O(N^2)$

题 73

试题编号	Codechef JULY 11
试题名称	Billboards

题目大意	算法讨论
有一个长度为 n 的 01 序列，它任意长度为 m 的连续子序列中都有 k 个 1。问所有满足条件的 01 序列中，1 的数目最少的不同的序列有多少个？ 数据范围：$1 \le k \le m \le 50, m \le n \le 10^9$	先考虑 m 整除 n 的情况，可以把串分成 $\frac{n}{m}$ 段，每一段的长度都是 m，那么每一段中至少有 k 个 1，整个串至少有 $\frac{kn}{m}$ 个 1，每一段的最后 k 个放 1 是满足条件的，所以 1 的个数最少是 $\frac{nk}{m}$。 现在对每一个序列构造一个 $k \times \frac{n}{m}$ 的矩阵 A，矩阵的第 i 行第 j 列表示的是第 j 段中第 i 个 1 在这一段中的位置（注意不是整个串中的位置），如果一个序列是合法的，那么就满足这个矩阵的每一行单调不增，每一列单调增。显然这样的矩阵和合法的方案一一对应。 这是一个半标准的杨氏矩阵，方案数的计算公式如下：$ans = \prod(i,j)r + j - ihook(i,j)$，其中 r 表示候选数集合的大小，$hook(i,j)$ 表示和 (i,j) 同一行或同一列且至少有一维坐标比它大的位置数。但是因为这个矩阵的坐标范围很大，所以直接计算会超时，如果把分子乘的数和分母乘的数列出来，可以发现大部分都是可以约分的，真正要计算的数的个数只有 $O(km)$，直接计算即可。 当 m 不整除 n 的时候，如果 $n \mod m \le m-k$，那么每一组的前 $n \mod m$ 个数都是 0，否则每一组的后 $m-n \mod m$ 个位置都是 1，把确定的部分去掉后就又转化成了 m 整除 n 的情况。

时空复杂度	时间复杂度 $O(mk)$，空间复杂度 $O(1)$

题 74

试题编号	Codechef JUNE 12
试题名称	Expected Maximum Matching

题目大意	算法讨论
给定一个 $n \times m$ 个矩阵 $f_{i,j}$，按照如下方式生成一个左边 n 个点右边 m 个点的二分图：左边第 i 个点和右边第 j 个点之间右边的概率为 $f_{i,j}$，求这样生成的二分图的最大匹配的期望值。 数据范围：$n \le 5, 1 \le m \le 100, 0 \le f_{i,j} \le 1$	根据 Hall 定理，如果左边的集合 S 可以完全和右边匹配，那么每一个 S 的子集都至少与 $\|S\|$ 个右边的点连接。 考虑状压 DP，令 $dp_{i,K}$ 表示只考虑右边的前 i 个点，左边的所有点的每一个子集满足上述条件的情况为 K。因为一共只有 2^5 个子集，所以 K 可以用一个 unsigned int 来存储。可以发现上述条件的满足情况存在包含关系，即如果 S 满足条件，那么 S 的每一个子集一定都满足条件，所以实际上可能的 K 是不多的，在 n 等于 5 的时候只有不到 4000 个，预处理出所有可能的 K 和转移，直接 DP 就行了。

时空复杂度	时间复杂度 $O(mP)$，空间复杂度 $O(mP)$，其中 P 为可能的 K 的个数

题 75

试题编号	Codechef FEB 15
试题名称	Payton numbers

题目大意	算法讨论
定义三元组 (a,b,c) 的乘法运算，其中 $c=11$ 或者 $c=24$。如果令 $(a1,b1,c1)$ 和 $(a2,b2,c2)$ 相乘： 令 $s=(a1a2+b1b2+c1c2)+(a1b2+b1a2)+(c1+c2), t=\lfloor\frac{s}{2}\rfloor+16(c1+c2)-c1c2$ $A=(t-2(a1b2+b1a2)-(a1c2-c1a2)+33(a1+a2)+(b1b2-a1a2))$ $B=(t-5(a1b2+b1a2)-(c1b2+b1c2)+33(b1+b2)+(2b1b2+4a1a2))$ 如果 s 是偶数，那么结果就是 $(A-540,B-540,24)$，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 $A\times B=B$ 的三元组，定义 $zeroA$ 是对任何 B 都满足 $A\times B=A$ 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，判断它是不是素数。 数据范围：$T\le 10^4, -10^7\le a,b\le 10^7$	令 ω 为满足方程 $\omega^2=\omega-3$ 的解，即 $w=\frac{1+\sqrt{-11}}{2}$，那么对于每一个三元组 (a,b,c), 都有到域 $Z[\omega]$ 的映射 $\phi(a,b,c)=(33-2a-c)+(b-a)\omega$。于是问题就转化为了判断域 $Z[\omega]$ 下的数 $a+b\omega$ 是否为素数。 定义共轭 $(a+b\omega)'=(a+b-b\omega)$，那么令 $Nx=xx'$，那么就有如下结论：(1) 如果 x 不是整数，那么 x 是质数当且仅当 Nx 是质数。(2) 如果 x 是整数，那么 x 是质数当且仅当 x 是质数且要么 $\|x\|=2$，要么 $\|x\|\neq 11$ 且 -11 在模 x 域下没有二次剩余。 于是可以直接用欧拉判别法和 miller rabin 来判断。

时空复杂度	时间复杂度 $O(T\log\|a\|)$，空间复杂度 $O(1)$

题 76

试题编号	Codechef FEB 13
试题名称	Room Corner

题目大意	算法讨论
给定一个 $n\times m$ 的网格图，用 $+,-,\|$ 描述了一个房间的边界，保证房间内的所有格子都是四联通的且同一行的所有房间内的格子相邻。现在这个房间所有 90 度的内角对应的格子中都站了一个小朋友，小朋友移动的时候把手放那个内角所对应的墙上，且每时每刻手不能离开墙壁。我们把 90 度内角所对应的空白格成为特殊点。 接着游戏开始，每时每刻两个相邻的（即之间没有任何小朋友）的都处在特殊点上的小朋友可以交换位置，它们都向对方的特殊点移动且这次交换直到双方同时到达对方的特殊点才终止。小朋友移动到相邻的空白格需要一个单位的时间。现在有 T 组询问，每组询问问在最优情况下两个小盆友相遇至少需要多少时间。 因为这题题面细节非常多所以这儿就大致口胡一下.. 至于更多的细节请看原题题面。 数据范围：$T\le 10^4$,$n,m\le 2500$	我们可以从第一行最左边的小朋友开始，以逆时针顺序遍历整个网格的边界，这样就得到了一个环，我们记录下来环上每一个关键点到出发点的距离，这一部分的细节很多。每一次交换肯定都是环上相邻的两个关键点进行了交换。 对两个小朋友来说，使得它们相遇尽可能早的最优情况一定是它们同时开始相向交换，直到某一时刻它们相邻。这时等到双方都完成当前的交换之后，它们再进行最后一次交换。因此我们分两种出发方向（因为是个环）分别考虑，每一次二分处它们路径的中点在哪两个小朋友之间，那么通过前缀和就可以得到当前的最优解了。把两种情况的最优解取较小值就是答案了。

时空复杂度	时间复杂度 $O(nm+T\log n)$，空间复杂度 $O(nm)$

题 77

试题编号	Codechef APRIL 15
试题名称	Little Party

题目大意	算法讨论
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 s 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 $s_i = A_{S_i}$。现在你需要使用一些基子集，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。你需要最小化使用的基子集的大小的和。 数据范围：$T \le 120$,$m \le 1000, n \le 5$	这个问题可以转化为最小带权集合覆盖问题，这是一个 NPC 问题。所以考虑对搜索进行优化，我们先枚举所有的基子集，判断它可不可行，然后把可行的存下来，可行的基子集的数目最多是 3^n，所以直接搜索的总的复杂度是 $O(2^{3^n})$，这是难以接受的。 如果一个基子集能被另一个基子集完全包含，那么这个基子集就没有必要选，这样就能去掉一些基子集，枚举所有情况可以发现这时最多有 32 个基子集。在搜索的过程中，如果当前总代价超过当前最优解，剩下的基子集即使全部选上都不能完全覆盖或者当前基子集不能覆盖更多的元素，那么都可以进行剪枝，用位运算优化一下搜索的过程，就能在时限内得到答案了。

时空复杂度	时间复杂度 $O(T2^{3^n})$，空间复杂度 $O(3^n + m)$

题 78

试题编号	Codechef AUG 11
试题名称	Something About Divisors

题目大意	算法讨论
对于给定的正整数 B 和 X，求满足条件的正整数 N 的个数: 要求对于 N，至少存在一个数 $D(N < D <= B)$ 能整除 $N \times X$。 数据范围：$T \le 40$,$X \le 60, B \le 10^{12}$	令正整数 $i = \frac{NX}{D}$，那么显然有 $i < X$，考虑枚举每一个可能的 i，为了避免重复的统计，我们可以计算满足 $i\|NX$ 且不存在 j 满足 $i < j < X$ 且 $j\|NX$ 的 N。 因为 $i\|N \times X$，所有有 $\frac{i}{\gcd(i,X)}\|N$，令 $A_i = \frac{i}{\gcd(i,X)}$，那么就有 $N = A_i p$。因为有 $N \le \frac{Bi}{X}$，所以就有 $p \le \frac{Bi}{XA_i}$，把这个取值的上界记为 P。 因为 $j\|A_iX \times p$，所以有 $\frac{j}{\gcd(A_iX,j)}\|M$，令 $B_j = \frac{j}{\gcd(A_iX,j)}$。那么满足 $i\|NX$ 的同时满足 $j\|NX$ 的条件是 $D_j\|M$。所以我们可以枚举 i，然后对所有 $i < j < X$，进行容斥，即 $ans_i = \sum(-1)^t \frac{P}{\text{lcm } B_j}$。 直接计算显然会超时，但是可以发现当分子大于 P 时贡献就是 0 了，再加上可能的 lcm 的值也不多，所以就可以用 DP 的方式来进行。注意优化一下常数就能通过了。

时空复杂度	时间复杂度 $O(X^2K)$，空间复杂度 $O(XK)$，其中 K 为可能的 lcm 数量，大约为 10^4

题 79

试题编号	Codechef SEP 12
试题名称	Annual Parade

题目大意	算法讨论
一张 n 个点 m 条边的带边权有向图。有 K 组询问，每组询问给出一个整数 C。 对于每组询问，你需要从图中选出若干条路径（不能是自环且相同的边可以经过多次），一个方案的代价是所有经过的道路（多次经过重复统计）的边权和加上起点不等于终点的路径条数乘 C 再加上没有经过的城市数目乘 C。 对每组询问你需要计算最少代价。 数据范围：$2 \le N \le 250, 1 \le M \le 3 \times 10^4, 1 \le K \le 10^4$	先跑 Floyd，然后对每一个城市拆点，左边第 i 个点到右边第 j 个点之间连上流量为 1，代价为 i 到 j 之间的最短路。对这张图跑费用流并把初始的代价设为 nC。可以发现每一次增广后，实际的代价都会加上流量并减去 C。对于每一个询问，C 是给定的，增广路都是固定的，所以先把所有增广路都预处理出来，然后读入 C 后二分一下输出答案就好了。

时空复杂度	时间复杂度 $O(n^3 + m + K\log n)$，空间复杂度 $O(n^2 + m)$

题 80

试题编号	Codechef JULY 11
试题名称	Trial of Doom

题目大意	算法讨论
给定一个 $n \times m$ 的网格，每一个格子为蓝色或者红色，一共有 k 个红色格子，你可以在这个网格里走，每当你离开一个格子的时候，这个格子和它的四周四个格子会改变颜色，现在让你判断是否存在一条左下角到右上角的可以经过相同格子的路径，使得从左下角开始，沿着这条路径走，走到右上角再离开这个网格后，网格里所有格子都变成了蓝色。 数据范围：$T \le 50, n, m \le 10^9, \min(n, m) \le 40, k \le 10000$	我们假定 $n \le m$，当 $n > 1$ 的时候，路径的限制是不重要的，即对于任意的格子的集合 S，都存在一条路径使得这条路径经过了 S 中的格子奇数次，其它格子偶数次。这样就转化成了一个经典问题，只要能够列出方程，就可以使用高斯消元来解决。可以发现此时每一个红色的格子都是独立的，只有格子 (i, j) 是红色，等价于只有格子 $(i-1, j-1), (i-1, j), (i-2, j), (i-1, j+1)$ 是红色的。因此可以使用这种方法把第 j 列的格子给移动到第 $j-1$ 列上。 因为 n 很小，可以发现一个第 i 列的格子对第 j 列和第 $j-1$ 列的影响是存在一个周期的。即存在一个数 T 使得对第 $j, j-1$ 列的影响和第 $j-T, j-T-1$ 列的影响是一样的。那么可以先预处理出这个循环节，那么就可以在 $O(nK)$ 的时间内把所有红色格子都移动到第一列。同样利用这个循环节，我们可以把设在最后一列的变量给移动到第一列，这样就可以得到 n 个方程，判断这个方程组是否有解即可。上述过程可以用位运算优化掉一个 n。 当 $n = 1$ 的时候，路径的长度的奇偶性和 n 一定相同，于是我们可以先暴力枚举第一个格子经过次数的奇偶性，然后解出每一个格子经过了多少次，然后分别判断一下和的奇偶性是否满足要求即可。

时空复杂度	时间复杂度 $O(Tk + TnLen)$，空间复杂度 $O(k + len)$，Len 为循环节长度

题 81

试题编号	Codechef APRIL 13
试题名称	String Query

题目大意	算法讨论
给定一个长度为 10 的只包含小写字母的字符串 S，有 n 个操作：1. 在首或者尾或者正中间插入一个小写字符。2. 在首或者尾或者正中间删除一个小写字符，保证字符串长度不会小于 10。3. 给定一个字符串 T，问 T 在 S 中出现了多少次。 数据范围：$n \le 1.5 \times 10^5$ 且询问串长度之和不超过 1.5×10^6	如果只有在一端插入或者删除，我们可以用传统的后缀平衡树来解决这个问题。如果只有在头部或者尾部插入或者删除，我们可以维护两棵后缀平衡树 L 和 R，并把它们作为一个整体 T，对于头部的操作都在 L 中进行，对于尾部的操作都在 R 中进行。如果出现删除时这一边的后缀平衡树中不存在节点了，我们暴力重构整个结构，即取出这个结构所代表的字符串，把两棵平衡树清空，把前半段插入 L，后半段插入 R。可以证明这样的均摊复杂度还是 $O(n \log n)$。至于在中间插入和删除，我们可以维护两个 T 结构 TL 和 TR，每一次插入之后在两个结构之间交换一些字符使得中间位置恰好在 TR 的头部。 至于查询，我们可以分三种情况考虑：1. 查询串的所有字符都在同一棵后缀平衡树中，那么我们可以在平衡树中暴力找出小于它的最大的后缀数量和不大于它最大的后缀数量，作差即可，这一部分的复杂度是 $O(\|T\| \log n)$。2. 所有字符在同一个结构 T 中单不在同一棵后缀平衡树中，这时我们取出 L 中的后 $\|T\|-1$ 个字符和 R 中的前 $\|T\|-1$ 个字符拼接在一起跑一遍 kmp。3. 所有字符不在同一个结构中，解决方法和 2 类似。

时空复杂度	时间复杂度 $O((n+\|S\|)\log n)$，空间复杂度 $O(n)$

题 82

试题编号	Codechef JUNE 12
试题名称	Cool Numbers

题目大意	算法讨论
一个数 A 有 k 位，从中选出至多 3 个不同的数位，令这几位的和为 S，这个数的所有数位和为 K，如果存在一种选取方案满足 $(K-S)^S$ 是 A 的倍数，那么就把 A 称为 cool number。现在给定一个 n，求小于等于 n 的最大的 cool number 和大于 n 的最小的 cool number。 数据范围：$T \le 10^5, n \le 10^{1000}$，保证所有数据中 n 的位数和不超过 4×10^6。	我们可以把 cool number 分成两类，第一类 cool number 只有不超过 3 个非 0 位，所有这样的数显然都是 cool number，大于一个数最小的和小于等于一个数最大的这一类数可以很方便的求得。于是接下来我们只考虑剩下的 cool number，即第二类 cool number。 对于第二类 cool number，满足 $(9k-27)^{27} > 10^{k-1}$，于是可以得到一个 k 的上界为 77。因此第二类的 cool number 是有限的，我们可以先预处理出所有第二类 cool number，然后询问的时候只需要二分就好了。至于预处理，可以枚举 $K-S$，然后枚举 $(K-S)^{27}$ 的所有不超过 10^{77} 的约数进行判定。可以发现第二类的 cool number 只有不到 4×10^4 个。但是直接暴力预处理会稍微有点超时，我们可以放宽一点条件，比如缩小一点 $K-S$ 的枚举范围或者 10^{77} 的限制，把剩下的数打表加入程序中，这样就能在时间限制内得到答案了。

时空复杂度	时间复杂度 $O(\text{预处理} + T(\log n + \log N))$，其中 N 为第二类数的个数，空间复杂度 $O(N)$

题 83

试题编号	Codechef JAN 12
试题名称	Misinterpretation 2
题目大意	算法讨论
一个长度为 n 的只包含小写字母的字符串 S，如果把它的偶数位依次写到开头，再把奇数位依次写下去，得到的字符串和原串一样，那么称这个字符串是好的。给定 L,R，问长度为在 L,R 之间的好的字符串有多少个。 数据范围：$T\leq 5, L,R\leq 10^{10}$,$R-L\leq 5\times 10^4$。	可以发现重排对应着一个置换，如果这个置换的循环数为 $f(n)$，那么长度为 n 的好的字符串个数就有 $26^{f(n)}$ 个。当 n 为偶数的时候，置换之后第 i 为在第 $2i \mod (n+1)$ 位，而当 n 为奇数的时候，最后一位不变，而前面的排列和 $n-1$ 的情况一样，所以此时有 $f(n)=f(n-1)+1$。于是只需要考虑 n 为偶数的情况。 令 $\mathrm{ord}(mo)$ 为 2 模 mo 的阶，可以发现所有 $\gcd(i+1,n)=p$ 的 i 每 $\mathrm{ord}(\frac{n+1}{p})$ 一组构成了若干个置换，而这样的数有 $\phi(\frac{n+1}{p})$ 个，所以可以得到 $f(n)=\sum_{p\mid(n+1),p>1}\frac{\phi(p)}{\mathrm{ord}(p)}$。 于是只需要考虑怎么求 $f(n)$ 即可。第一步是求所有 n 的约数，我们可以对所有 $n\in[L-1,R+1]$ 进行分解质因数，因为这些数是连续的，所以我们可以预处理出 1 到 $\sqrt{R}$ 的所有质数然后对这个范围内的数依次除过去，得到质因数分解之后就能简单地得到所有约数了。第二步是求 $\mathrm{ord}(p)$，可以发现当 a,b 互质的时候，有 $\mathrm{ord}(ab)=\mathrm{lcm}(\mathrm{ord}(a),\mathrm{ord}(b))$。所以只需要对质数 $ord(p)$ 即可，这时我们可以直接枚举 $p-1$ 的约数一一检查来得到答案，当 $p\leq 10^5$ 的时候我们可以直接预处理出来，这样对于每一个 $n\in[L-1,R+1]$，至多只有一个质数要重新计算。之后直接按照式子累加答案即可。
时空复杂度	时间复杂度 $O(\frac{R^{\frac{3}{4}}}{\log R}+\sqrt{R}\log R+T(R-L)(\frac{\sqrt{R}}{\log R}+\log^2 R))$，空间复杂度 $O(\sqrt{R}+(R-L)\log R)$

题 84

试题编号	Codechef OCT 14
试题名称	Union on Tree
题目大意	算法讨论
给定一棵 n 个节点的树，树的每一条边的长度都是 1。接下来有 Q 组询问，每组询问给出了 K_i 个数对 (a_i,r_i)，表示距离点 a_i 在 r_i 以内的所有点都被守护了，问一共有多少点被守护了。 数据范围：$n,Q\leq 5\times 10^4$ $\sum K_i\leq 5\times 10^5$。	如果 $K_i=1$，那么我们可以通过点分治预处理来得到答案，我们可以对每一个可能的 k，预处理出点分治第 i 层，每一个中心 j 所控制的范围内距离它小于等于 k 的点的个数。同时为了避免重复计数，我们令第 $i-1$ 层分治中 j 的父亲为 f_j，还要对每一个可能的 k 预处理出第 i 中中心 j 所控制的范围内距离 f_j 小于等于 k 的点的个数。那么在统计的时候我们从这个点开始依次访问点分结构中的父亲累加答案即可（还要预处理每一个 j 到 f_j 的距离）。 当 $K>1$ 的时候，我们可以对给定的关键点建出虚树。考虑虚树上的点 a 和 b，如果它们的距离为 d，且 b 的守护范围为 c，那么 b 给 a 的贡献就是 $c-d$，如果 a 的守护距离小于这个值，那么 b 的实际的守护距离就可以看成这个值。因此我们可以得到一个类似最短路的算法来求出虚树上每一个节点的守护距离。最后，我们分虚树上的每一条边统计，对于虚树上的一条边，一定存在一个点 u 使得两个端点对它的贡献相同，我们把它的守护距离设为这个贡献。可以证明答案就是所有端点能够守护到的点的个数减去所有中间点能到达的点的个数。
时空复杂度	时间复杂度 $O(n\log n+\sum K\log n)$，空间复杂度 $O(n\log n)$

题 85

试题编号	Codechef DEC 14
试题名称	Divide or die

题目大意	算法讨论
平面上给定一个 n 度角，你需要通过尺规作图来把这个角分割成 n 的大小为 1 度的角。你的操作有：1. 以 A, B 为给顶点画一条直线。2. 以 A 为圆心，B, C 之间距离为半径话一个圆。你用到的点只能是最开始给定的三个点或者你绘制的图形之间的交点。 数据范围：$0 < n < 360$, 坐标范围不超过 1000，你的操作次数不能超过 1000 次。	可以发现 $n \mod 3 = 0$ 的时候无解，其它情况都是有解的。我们可以先绘制一个正五边形，这样就得到了一个 72 度角。接着我们绘制一个正三角形，这样就得到了一个 60 度角。作差之后就得到了一个 12 度角。平分两次就得到了一个 3 度角。我们不停用原来的角减去这个 3 度角。最后就得到了一个 1 度角或者 2 度角，如果是 2 度角就再平分一次依然能够得到一个 1 度角。最后用不停地用原角减去这个 1 度角，就能把原角 n 等分了。

时空复杂度	时间复杂度 $O(n)$，空间复杂度 $O(n)$

题 86

试题编号	Codechef JUNE 11
试题名称	Minesweeper Reversed

题目大意	算法讨论
给定你一个 $R \times C$ 的扫雷棋盘，其中的雷的位置已经表明。最开始所有的方块都是打开的，你需要关闭所有的方块。你可以通过一次点击来关闭一个方块（可以关闭含雷的方块）。在你关闭 (x, y) 后，在正常的扫雷游戏中可能和 (x, y) 同时被打开的格子都会被关闭。现在要你求出至少点击多少次，可以关闭所有的方块。 数据范围：$T \leq 50, 1\ leqR, C \leq 50$	显然每一个雷都要花费一次点击来关闭。我们把剩下的格子分成两类，第一类是和雷相邻的，第二类是不和雷相邻的。其中第二类格子构成了若干个联通块（注意这儿的相邻和联通都是八连通意义下的）。那么在正常的扫雷中，如果打开了一个联通块，所有和它相邻的第一类格子都会被打开。所以在这儿如果我关闭了一个第一类格子，那么所有和它相邻的联通块以及和这些联通块相邻的第一类格子都会被关闭。 可以发现第一类格子至多与两个联通块相邻，如果不与任何联通块相邻，那么它也必须消耗一次点击。把和两个联通块相邻的第一类格子看成边，联通块看成点。那么关闭所有联通块的代价就是联通块数减去这张无向图的匹配数。使用带花树即可。

时空复杂度	时间复杂度 $O(TR^2C^2)$，空间复杂度 $O(RC)$

题 87

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

题目大意	算法讨论
给定一张无向图，两个人玩游戏。第一个选择一个出发点，接着第二个人开始两个玩家轮流操作，每一次操作可以沿着边移动到一个未被到达过的点，无法移动的人会输掉这个游戏。两个人都以最优策略移动，问有多少个出发点对第一个人来说是必胜的。 数据范围：$T \leq 3, n \leq 2000, m \leq 3 \times 10^5$	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。于是我们可以先用带花树求一遍最大匹配。可以发现和当前的任意孤立点之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出发再做一次增广，把所有距离为偶数或者花中的节点都标记为必胜。最后统计点数就是答案了。

时空复杂度	时间复杂度 $O(Tnm)$，空间复杂度 $O(m)$

题 88

试题编号	Codechef NOV 13
试题名称	Queries With Points

题目大意	算法讨论
平面上给出 n 个简单 K_i 边形。接着在线的给出 Q 次询问，每次询问给出一个点，你需要求出这个点在哪个多边形内（或者不在任何多边形内）。 数据范围：$n \le 10^5, \sum K_i \le 3 \times 10^5, Q \le 10^5$, 坐标范围不超过 10^9	如果没有强制在线，那么就是一个典型的点定位问题。我们使用扫描线算法，把所有顶点所在的 x 坐标设为关键点，那么任意一对相邻的关键点之间的线段数量，线段的 y 坐标顺序都是不变的。我们用平衡树来维护这些线段的 y 坐标顺序，然后对每一个查询点求出它上方的线段和下方的线段，判断一下两个线段是不是属于同一个多边形以及上方的是否是上边界，下方的是否是下边界即可。 因为这题有强制在线，所以我们可以用可持久化平衡树来记下所有情况写的平衡树形态来实现在线的查询。

时空复杂度	时间复杂度 $O(K \log K + Q \log K)$，空间复杂度 $O(K \log K)$

题 89

试题编号	Codechef DEC 11
试题名称	Hypertrees

题目大意	算法讨论
一个 3-超图类似与一个普通的图, 只不过其中的边都连接三个点. 一个 3-超树是一个去掉任意一条边都以后都不连通的 3-超图. 给定 n, 问有几种含有 n 个带标号的点的本质不同的 3-超树. 数据范围：$n \le 17$，保证答案小于 2^{63}	这是一道打表题。对于一棵超树，我们可以把它分成若干个点双联通分量，即对于每一个分量都有去掉任意一个点超树依然联通。可以发现对于一棵点双联通的超树，每一条边连接的三个点中都恰好有一个点为叶子节点（点数为 3 的情况除外）。我们把那个节点删掉，这样就得到了一个常规的点双联通图，原来的超图的点数对应着这个图的点数 + 边数。 我们可以用暴力的搜索求出点数为 i 的双联通超树数量。接着我们暴力枚举这个超树的每一个点双联通分量的大小，然后使用记忆化搜索的方法求出这些点双联通的超树拼接而成的不同的超树数量。这样的程序在本地需要 5s 左右求出这 17 个输入的答案。把表提交即可。

时空复杂度	时间复杂度 $O(T)$，空间复杂度 $O(n)$

题 90

试题编号	Codechef OCT 13
试题名称	Three-Degree-Bounded Maximum Cost Subtree
题目大意	算法讨论
给定一张 n 个点 m 条边的无自环无重边的带边权的联通无向图，保证这个无向图的每一个点双联通分量大小都不超过 9。 现在你需要求出这张图满足每一个点的度数都不超过 3 的 K 个点 $K-1$ 条边的联通子图中边权和最大是多少，以及满足边权和最大的不同子图个数对 2^{32} 取模后的值。 数据范围：$T \le 6, n \le 100, m \le 450$，权值范围 10^4	因为这张图每一个点双联通分量都很小，所以我们可以先对这张图进行点双的缩点。接着对每一个点双进行考虑，此时 $n \le 9$，我们可以用一个 DP 来求出答案，令 f_i 为所有点的度数情况为 i 的时候，把所有度数不为 0 的点联通的满足条件的子图的最大边权和，w_i 为方案数。转移可以枚举编号最小的度数不为 0 的节点，如果它的度数为 1，那么可以枚举它和哪个点相邻，否则可以枚举它其中的一颗子树，这样就能规约到 i 更小的问题，就可以 DP 了。 接着在点分树上，所有节点都是割点或者点双联通分量。对一个点双联通分量，我们用它的父节点（必定是割点）来代表它。令 $A_{i,j}$ 为树上点 i 的代表点的度数为 j 时的最大值，$B_{i,j}$ 为方案数。可以通过一个简单的树形 DP 来得到答案。要注意统计的时候不能重复也不能遗漏，细节还是非常多的。
时空复杂度	时间复杂度 $O(T(m+n12^9))$，空间复杂度 $O(n4^9)$，但是因为大部分的状态都是无用状态，通过预处理去掉这些状态后跑的还是非常快的

题 91

试题编号	Codechef NOV 11
试题名称	Colored Domino Tilings and Cutsontest
题目大意	算法讨论
一个 N 行 M 列的矩形棋盘。一个棋盘覆盖的染色是指：在棋盘上填上小写字母，使得每个格子有且仅有一个相邻格子的字母和他的一样。一个格子与另一个格子相邻当且仅当他们有公共边。每个字母对应一种颜色。棋盘的割是指一条竖直或水平的直线将棋盘分成两半，这条直线不能穿过一对有相同颜色的相邻格子。 现在你需要构造一个在割数尽可能小的情况下，染色数尽可能小的染色。 数据范围：$T \le 3000, N, M \le 500$	无解当且仅当 NM 是奇数。让 N, M 都比较大的时候，显然无法用两种颜色染色，而我们可以可以构造出 6×8 和 5×6 的两种棋盘，这两个棋盘的割数是 0，染色数是 3，而且这两种棋盘的特点是可以在不改变割数的情况下增加两行或者两列（只需要添加一排横着的或者竖着的 1×2 单位即可）。于是我们可以通过这两种基础的网格构造出 N, M 比较大的时候割数等于 0 的解。至于染色数，可以发现构造出来的棋盘依然是满足三染色的，可以在拓展棋盘的过程中完成构造。 当 NM 比较小的时候我们可以通过打表来输出答案。
时空复杂度	时间复杂度 $O(TNM)$，空间复杂度 $O(NM)$。

以上是我做的 91 道常规题的题解，接下来是 challenge 型试题的题解。

题 92

<table>
<tr><td>试题编号</td><td colspan="2">Codechef APRIL 14</td></tr>
<tr><td>试题名称</td><td colspan="2">Sereja and Permutation</td></tr>
<tr><td colspan="2">题目大意</td><td>算法讨论</td></tr>
<tr><td colspan="2">给定一个长度为 n 数组 A' 和整数 S，你需要给这个数组重新排列，设你排列得到的数组为 A。
令函数 $f(A,i)=S-\sum_{k=i}^{j}A_k$，其中 j 是使得 $\sum_{k=i}^{j}A_k \le S$ 的最大整数，若 $A_i > S$，则 $f(A,i)=S$。
现在给定整数 k，你需要最小化 $\frac{\sum_{i=1}^{k}f(A,i)}{k}$
数据范围：$T \le 10, n \le 2000, A'_i \le 10^4$</td><td>考虑一种特殊情况，当 $S \ge \sum_{i=1}^{n}A'_i$ 的时候，显然 A 升序最优。可以发现，把 A' 从小到大排列对于普通的情况也是一种比较优秀的解，其实光是这么输出在 CC 上就有比较高的分数了。
考虑另外一种贪心方法。假设我们已经知道了第 $k+1$ 个数到第 n 个数是什么，我们通过排列剩下的数，最小化 $f(A,k)$，并在此基础上最小化 $f(A,K-1)$，依次类推。这样的贪心如果朴素的时限的话是 $O(n^2)$ 的，但是我们可以使用平衡树来优化这个过程到每次 $O(n\log n)$。
在这个复杂度下，时限范围内允许我们对 300 组左右的右侧选取方案进行贪心。于是我们可以随机 300 种右侧的选取方案（当然其中一部分可以加一些其他的限制，比如保证右侧的数递增或者递减之类的），分别贪心之后取最优的那个解即可。</td></tr>
<tr><td>时空复杂度</td><td colspan="2">时间复杂度 $O(TPn\log n)$，空间复杂度 $O(n)$，其中 P 是随机的组数。</td></tr>
</table>

题 93

<table>
<tr><td>试题编号</td><td colspan="2">Codechef JUNE 13</td></tr>
<tr><td>试题名称</td><td colspan="2">To challenge or not</td></tr>
<tr><td colspan="2">题目大意</td><td>算法讨论</td></tr>
<tr><td colspan="2">给定一个长度为 m 的数组 B，你需要从中选出尽可能多的数使得选出的数中不存在任意的三个数可以组成等差数列。
数据生成方式为随机一个 L 和 p，接着循环 1 到 L，每一个数都有 p 的概率加入 B 中。
数据范围：$10^4 \le L \le 10^5, 0.1 \le p \le 0.9$</td><td>考虑最简单的贪心，把数组排序之后，从小到大判断，如果当前的数能加入到选取的集合中，那么就选取这个数，否则就不取。假设这样得到的答案个数为 K，那么做一遍的复杂度是 $O(m+K^2)$ 的，因为这个做法得到的 K 不会很大，所以可以比较快地跑出来。
实际上直接这么贪心就可以得到比较高的分数，为了得到更优的解，我们可以采取一些优化的措施，比如当一个数能选取的时候，我们有 w 的概率不取它，并随机若干次取最优解。考虑到所有数据，可以选取 $w=\frac{1}{70}$。
而实际上，对所有数据 $w=\frac{1}{70}$ 并不都是最适合的，我们可以先采取一些方法得出一个比较优秀的 w，然后再进行上述的随机策略，就可以得到一个更好的解了，在我的程序中采取的方法是三分，使用的估价是对当前的 w 随机 T 次取最优值。T 的选取以及最后随机的次数可以对各类数据打一张表。</td></tr>
<tr><td>时空复杂度</td><td colspan="2">时间复杂度 $O(L(m+K^2))$，空间复杂度 $O(m)$，其中 L 是随机的组数，K 是答案。</td></tr>
</table>

题 94

试题编号	Codechef AUG 13
试题名称	Deleting numbers

题目大意	算法讨论
给定一个长度为 n 的数组 A，你需要对它进行若干轮的操作，每一轮你可以选取两个数 v 和 t，满足 $v+t \leq n-1$，且令 k 为最大的满足 $v+kt \leq n$ 的数，有 $A_v = A_{v+t} = ... = A_{v+kt}$。然后这些数将会被删除，以后形成的 $n-k$ 个数相对位置不变地形成新的数组 A，且新的 n 为 $n-k$。你需要用尽可能少地步数删除完所有的数。 数据范围：$n, A_i \leq 10^5$	这道题的操作比较复杂，因为在删除完数之后剩下的数的下标会产生变化，导致一些常用的策略比如 DP 难以发挥作用。但是如果我们每一次操作的都是数组 A 中比较靠后的数，那么就可以尽可能地减少这个的影响。 最简单的想法就是从后往前依次删掉所有的数，这样就能在 n 次操作内得到最优解。稍微优化一下，如果数组中出现次数最多的数是 K，那么我们可以从后往前删掉所有除 K 以外的数，然后用一次操作删掉所有数。 然而这些方法的效果都不是很好。考虑一种贪心的方法：每次我们找出最长的可以一次删除的数列，然后把它删除。这个做法每次操作的时间复杂度是 $O(n^2 \log n)$，而最坏情况下要进行 $O(n)$ 次，所以时间复杂度是 $O(n^3 \log n)$，尽管常数很小但是还是不可能再时间限制内得到答案。 于是我们可以把长度为 n 的数组分成若干段长度为 S 的小段，如果从后往前考虑消完所有的小段那么每一个小段都是一个独立的子问题，分别解决即可。至于 S 的选取可以根据 n 来打出一张表。用这种方法得到的解就已经比较优秀了。

时空复杂度	时间复杂度 $O(nS^2 \log S)$，空间复杂度 $O(n)$。

题 95

试题编号	Codechef JUNE 12
试题名称	Closest Points

题目大意	算法讨论
三维空间中有 n 个点，现在给出了 q 组询问，每一组询问给出了一个点，你需要从给定的 n 个点中找出距离这个点最近的点的编号。 数据范围：$n, q \leq 5 \times 10^5$，坐标范围不超过 10^9	这道题是一个经典问题，我们可以直接用传统的 kd-tree 来达到 100% 的正确率。 划分的时候要注意选取方差最大的那维来划分不然会 T，还有一点要注意的是坐标范围是 10^9，所以两点之间距离的平方可以达到 1.2×10^{19}，这已经超过 long long 的范围了。

时空复杂度	时间复杂度 $O(qn^{\frac{2}{3}})$，空间复杂度 $O(n)$。

题 96

试题编号	Codechef SEPT 12
试题名称	Simultaneous Nim

题目大意	算法讨论
给定 n 个异或和为 0 的数 A_i，你需要把它们分成尽可能多组，使得每一组的异或和都为 0。 数据范围：$n \leq 1000, A_i < 2^{60}$	考虑一个贪心的方法，每一次选取一个尽可能小的包含最小的那个数的集合使得这个集合的异或和为 0，然后把这个集合的数删去。于是问题就转化为如何找到一个集合的数使得它们的异或和为 0。 我们可以先把最小的那个数设为必选，对剩下的每一个数设一个 01 变量表示它是否要选，这样根据每一位的异或和都是 0 这一条件就可以列出 m 个方程，这个方程的每一组解就是一个合法的方案。 在这个方程组中，变量的个数有 $O(n)$ 个，而方程只有 m 个。为了使得解中 1 的个数尽可能的少，我们可以减少一些变量的个数，每一次我们只随机其中的 S 个数出来，用高斯消元判断它有没有解。现在当 $S = m$ 的时候这个方程就已经可能有解了。我们随机若干次，取 1 的个数最小的那个解作为当前选取的集合就可以了。当然如果当前的 S 随机了若干次之后依然没有解，我们可以略微增大 S 的值直到有解为止。 消元的过程可以用 bitset 优化。

时空复杂度	时间复杂度 $O(Tnm)$，空间复杂度 $O(n)$，T 表示每一个 S 的随机次数。

题 97

试题编号	Codechef OCT 11
试题名称	The Great Plain

题目大意	算法讨论
给定一个 $n \times m$ 的网格，网格中有一些格子已经填上了 1 到 50 的整数，你需要在其他的格子中填上这个范围内的整数。 填好之后，网格中任意一对相邻的格子 (i, j) 会产生 $2^{\lvert A_i - A_j \rvert}$ 的代价，其中 A_i 表示格子 i 上的整数，两个格子时相邻的当且仅当它们恰好存在一条公共边。 你需要尽可能地最小化网格的代价和。 数据范围：$T \leq 10, n, m$ 在 $[10, 100]$ 中随机生成。	考虑一个贪心的方法，我们扫描每一个网格，在其它格子都不变的情况下调整它的值，使得它和它周围的格子产生的代价和最低。显然这是一个迭代的过程，即扫描完网格之后我们再扫描一次，格子的值依然有可能发生变化。我们可以扫描若干次直到它稳定为止。 上述做法的效果依赖于初始解的选择，我们可以用贪心的方法来选择初始解，即把每一个空白格子划分给离它曼哈顿距离最近的初始给定的格子，并把它的权值赋为那个格子的权值。 这个做法存在一个弊端，就是在迭代的过程中可能会陷入一个比较劣的解的死锁中，即每一个格子都没有办法进行修改，但是这个解很差。因此随机的扰动是比较必要的，可以在每一次迭代之后都随机 S 个格子把它们的权值减去 1，在我的程序中选取了 $S = \frac{nm}{60}$。 还有一个优化是，与空白格子产生的代价和初始给定的格子产生的代价的估价应该是不同的，因为空白格子可以继续调整而初始给定的格子不行。因此在估价的时候可以给它们分别乘上一个常数，在我的程序中我选取的比例是 $6:7$。 通过这种方法就能得到一个比较优秀的解了。

时空复杂度	时间复杂度 $O(TPnm)$，空间复杂度 $O(nm)$，P 表示随机次数。

题 98

<table>
<tr><td>试题编号</td><td>Codechef JULY 11</td></tr>
<tr><td>试题名称</td><td>Large Kitchen</td></tr>
<tr><td>题目大意</td><td>算法讨论</td></tr>
<tr><td>给定一个 $n \times m$ 的网格，你需要在其中填上黑色格子，使得所有黑色格子时四联通的且黑格子之间不存在四联通的回路。你需要尽可能地最大化黑色格子的数量。
数据范围：$T \le 30, n, m \le 100$。</td><td>我们可以尝试得到一个局部的最优解，即当网格是 7×6 的时候，以下的网格一定是最优的

******
..*.
**.*.*
.*****
**.*.*
..*.
******

而其他的情况，我们可以把这个网格拓展开去，m 的拓展很贱但，直接延生即可，n 在拓展的时候要注意有些时候黑色格子会不连通或者可以调整黑色格子的位置使得个数变多。这个时候需要分一些情况考虑进行特判。
还有一点要注意的是通过上述做法构造出的 $n \times m$ 和 $m \times n$ 的答案可能不同，需要取最优值。</td></tr>
<tr><td>时空复杂度</td><td>时间复杂度 $O(Tnm)$，空间复杂度 $O(nm)$。</td></tr>
</table>

题 99

<table>
<tr><td>试题编号</td><td>Codechef NOV 14</td></tr>
<tr><td>试题名称</td><td>The Spelling Problem</td></tr>
<tr><td>题目大意</td><td>算法讨论</td></tr>
<tr><td>给定一篇英语文章，和一个字典，保证所有可能正确的英语单词都出现在了这个字典中。但是这篇英语文章在录入的时候出现了一些打字的错误，可能出现的错误有：交换了两个字符，漏了一个字符，多了一个字符，错了一个字符。
你需要尽可能多地找出文章中的错误并把它改正。把原来错的改对可以得 3 分，把原来对的改错会被扣 1 分，初始分数是 100。你要尽可能地最大化你的得分。
数据范围：文本大小不超过 10M，字典给定。</td><td>题目中给出的字典非常大，但是其中有很多是在交流中不可能出现的奇葩单词。所以我们可以百度一个牛津 3000 来用，我们可以对输入中的每一个词设置一个估价，估价可以设置三个常数 A, B, C，如果这个词在牛津 3000 中出现了，估计就增加 A；在题目中给定的词典中出现了，估价就增加 B；在文本串中每出现一次就增加 C。
之后对于每一个单词，如果它的估价超过了一个阈值，那么就把它视为是正确的。否则我们枚举每一种可能的错误方式，并枚举每一种可能通过这个错误方式得到当前单词的所有可能的单词，找出估价最大的作为一个备选方案。
这样我们就得到了 4 个备选答案，因为每一种错误的方法出现的概率是不同的，所以可以对每一种错误方法设一个估价，并选取估价最大的作为这个单词正确值。
具体的匹配过程可以用哈希和 Trie 树来实现。因为输入文本很大，全部处理会 TLE，所以可以仅截取前面一段来求出答案。</td></tr>
<tr><td>时空复杂度</td><td>时间复杂度 $O(NL^2)$，空间复杂度 $O(NL)$，其中 L 是单词的长度，N 是你截取的一段的长度。</td></tr>
</table>

题 100

试题编号	Codechef DEC 12		
试题名称	WordNinjas		
题目大意		算法讨论	
这一题的题面非常长，这儿仅作简要概括.. 有 10^4 个方块，它们被分入了 n 个集合，每一个集合的大小在 1 到 10 之间。方块有两种，分别为字母方块和特殊方块。特殊方块里又有各种各样的加分分块和空白方块。空白方块可以当做任何字母来使用。 你按照编号依次访问每一个集合，对于每一个集合，你可以取出其中的一个方块或者不选，如果你手上已经有 7 个方块了，那么取出的字母方块和空白放苦啊将会被直接丢弃。接着你可以选择丢掉一些你手上的方块。最后你可以选择是否用你手上的方块拼出一个单词。如果拼出了一个在字典中的单词，那么你就可以得到相应的得分，且你手上除了这个单词最后一个方块以外的所有方块都会被丢弃。 你需要尽可能地最大化你的得分。 数据范围：字典给定，大小约为 70000，字典中每一个单词的长度都在 2 到 7 之间，数据以某种非常科学的方式随机，你的总得分在 15000 以上就算通过。		观察题目中的算分方式，可以发现长度为 7 的字符串的评分非常高，光是加分就有 42 分。因为数据随机，所以输入的集合数量大约有 2000 个，如果每 10 个拼成一个长度为 7 的单词，那么仅仅是加分就有 8400 分，再加上字母单独的得分，到达 15000 还是绰绰有余的。 所以我们可以先不管加分方块和长度不为 7 的单词，以最大化长度为 7 的单词个数为首要目标。我们可以使用贪心的方法，维护一个指针 l，并用变量 r 遍历所有的集合，如果下标在 $[l, r]$ 的集合能够拼出一个单词，那么就把这个单词加入答案，然后令 $l = r + 1$。 那么现在要解决的问题就是如何判断一个单词能否被拼出。我们可以发现这可以转化为一个二分图匹配的问题，如果集合 i 中可以取出一个方块来表示单词的第 j 位，就在左边第 i 个节点到右边第 j 个节点之间连上一条边。如果这个二分图的匹配数等于 7，那么说明这个区间的集合可以拼出这个单词。拼出单词之后，我们可以访问没有匹配的块，看它里面有没有加分方块，如果有就无脑选上。 使用这个方法，就能够在清澄上通过这题了。	
时空复杂度	时间复杂度 $O(nNL^3)$，空间复杂度 $O(NL)$，其中 N 是长度为 7 的单词的个数，大约为 30000，$L = 7$，但是因为这个算法的常数非常小，还是可以在时限内跑出来的		

题 101

<table>
<tr><td>试题编号</td><td colspan="2">Codechef MAY 14</td></tr>
<tr><td>试题名称</td><td colspan="2">The Malaysian Flight Search</td></tr>
<tr><td colspan="2">题目大意</td><td>算法讨论</td></tr>
<tr><td colspan="2">这是一道交互题，你需要搜索马航的黑盒子。我们可以把地图简化为一个二维网格图。你需要确定黑盒子在哪一个网格中。现在我们确定了一个搜索范围，这个搜索范围是一个四联通的联通块，现在给定了这个四联通块的边界，这个边界有 n 个点。
接着你有 m 架飞机，每一个飞机有两个属性 R 和 C。每一次你可以把一架飞机派向搜索范围内的一个格子 (x, y)，假设黑盒子所在的网格为 $(x1, y1)$，如果有 $|x1 - x| + |y1 - y| \leq R_i$，那么返回值就是 1，否则就是 0。这个操作要付出的代价是 $C_i(|x|+|y|)$。保证一定存在 $R = 0$ 的点。
现在你需要找出黑盒子的位置，注意黑盒子可能不在搜索范围内，这个时候要返回 $(-1, -1)$。
数据范围：$n \leq 5000, m \leq 50$, 搜索范围在一个 1000×1000 的网格内，搜索范围内的整点个数不超过 2×10^5，坐标范围不超过 10^5</td><td>一个可行解是，我们直接用 $R = 0$ 的飞机访问搜索范围内的所有点，如果有返回值为 1，那么就得到了答案，于是要解决的问题就是如何找到搜索范围内的所有点。
因为坐标范围在一个 1000×1000 的网格中，所以我们可以把坐标范围给缩小至 1000。接着我们从一个一定不再范围内的点开始 BFS，如果碰到题目中给定的边界点，那么就跳出来，否则就把当前点染成黑色并继续 BFS，这样得到的白色点就都是搜索范围内的点了。
这样我们就得到了一个一定可以找到黑盒子位置的方法，但是这个做法非常劣，在清澄上得分只有不到 1 分。我们可以把其他的飞机利用起来。假设 $R = 0$ 的飞机的代价是 Q。我们按照 R 的值从大到小使用飞机，如果当前飞机满足 $C < Q(2R^2 + 1)$，那么我们就假定这个飞机是有优化的，我们用这个飞机依次去访问当前搜索范围内还没有被访问过的点，如果返回值是 1，那么我们就把搜索范围缩小到了 $O(R^2)$；否则我们就把同时在当前飞机的响应范围内和搜索范围内的点标记为访问过。通过这种方法，就能在刚才的方法的基础上得到比较大的优化。就能在清澄上通过了。</td></tr>
<tr><td>时空复杂度</td><td colspan="2">时间复杂度 $O(mNM)$，空间复杂度 $O(NM)$，N, M 为坐标范围</td></tr>
</table>