《IOI2015 作业做题报告》

绍兴一中王鉴浩

目 录

1 Rhombus

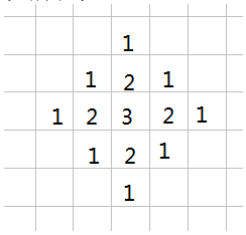
题目大意

有一个 $n*m$ 的矩阵，每个矩阵有一个值，需要选一个点 (x,y), $k \le x \le n-k+1, k \le y \le m-k+1$,设 $f_{x,y}=\sum_{i=1}^{n}\sum_{j=1}^{m} a_{i,j}*max(0,k-|i-x|-|j-y|)$ 求最大的 $f_{x,y}$。

数据范围：$1 \le n,m \le 1000, 1 \le k \le \lfloor \frac{min(n,m)+1}{2} \rfloor, 0 \le a_{i,j} \le 10^6$。

题解

我们把 $f_{x,y}$ 的公式分析，在 (x,y) 这个点的权值为 k，对整张图的贡献如下图所示：



由于有 $k \le x \le n-k+1, k \le y \le m-k+1$ 的限制存在，所以使得上图的贡献图是一定完整存在的。

我们设 x 轴正方向是向右的， y 轴正方向是向上的。

然后我们可以先算出 $f_{k,k}$ 的答案。

然后我们先考虑算出 $f_{x,k}$ 的答案再算出所有 $f()$ 的答案。

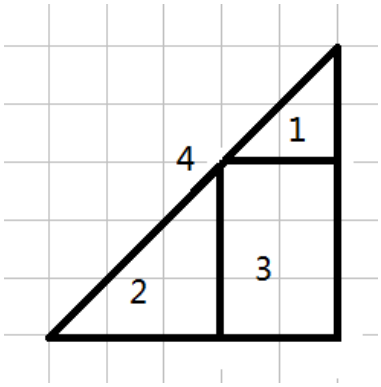
对于 $f_{x,y}$ 转移到 $f_{x+1,y}$ 的时候我们可以发现这个图形的左半个全部要减一的权值，对于新的图形的右半个全部要加一的权值。

而 $f_{x,y}$ 转移到 $f_{x,y+1}$ 则类似。

那么现在我们需要计算半个图形的和。但我们发现计算半个图形的和比较难处理。

那么我们可以把半个图形再分成两个方向不同的直角三角形的和，直角边与坐标轴平行。

对于一个需求的三角形，我们可以如下图切割：



4 号表示整个三角形。

可以发现 $1 = 4 - 2 - 3$，而 $4, 2, 3$ 都可以通过容斥预处理计算得到。

那么我们转移便可以 $O(1)$ 完成。

此题**时间复杂度为 $O(n * m)$，空间复杂度为 $O(n * m)$**。

来源

CF 263E

2 Distinct Paths

题目大意

一个 $n*m$ 的矩形，一开始一些格子被涂上了颜色，颜色只有 k 种。

问有多少种把未涂色格子涂色的方案使得从左上角到右下角的每条路径都不会经过两个颜色一样的块。路径只能向右或向下走。

输出答案模 1000000007。

数据范围：$1 \leq n, m \leq 1000, 1 \leq k \leq 10$。

题解

我们首先可以发现当 $n+m-1 > k$ 的话就是无解，因为路径长度就比 k 大了。

由于 n,m,k 都很小，我们可以假设暴力是可以工作的。现在我们要尽量缩减状态。那么我们可以用某种顺序枚举格子，我们可以把颜色重新编号，使得每个颜色第一次出现的时刻顺序一定是与搜的顺序符合的。

然后我们计算答案的时候只需要再乘一个组合数就可以了。

我们用暴力验证可以发现最坏情况只有 100 万左右的状态数。

那么我们就可以直接搜出全部答案了。

时间复杂度 $O(100w*k)$，空间复杂度 $O(n*m)$。

来源

CF 293B

3 Number Challenge

题目大意

定义 $d(n)$ 为 n 的约数个数。给出 a, b, c 求 $\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{c} d(i*j*k)$。

输出答案模 1073741824(2^{30})。

数据范围：$1 \leq n \leq 2000$

题解

此题解法有两种。

第一种为：

约数个数等于每个不同的质因子的个数加一相乘。

那么我们可以把一个一个质数分开考虑。

$dp[i, j, k, p]$ 表示现在这个数是 $i*j*k$，里面的质因子没有小于 p 的，的约数个数。

然后我们可以枚举 i, j, k 中关于 p 的幂次是多少，然后进行转移。

注意到如果 $p^2 > max(i, j, k)$ 的话，i, j, k 都为质数或 1，那么可以直接容斥计算答案。

然后我们可以发现 $\lfloor \frac{a}{x} \rfloor$ 可能的结果只有 $\sqrt{a}$ 个，设 $n = max(a, b, c)$，**空间复杂度为** $O(n^{1.5} * \sqrt{}(n)/log)$。

转移的时候复杂度是 $log(p)^3$，那么这个算法的时间复杂度我们可以近似看作是**时间复杂度为** $O(n^2)$。

第二种为：

我们可以证明一个公式：

$\sum_{i\leq a, j\leq b, k\leq c} d(i, j, k) = \sum_{i\leq a, j\leq b, k\leq c, gcd(i,j)=gcd(i,k)=gcd(j,k)=1} \lfloor \frac{a}{i} \rfloor \lfloor \frac{b}{j} \rfloor \lfloor \frac{c}{k} \rfloor$

证明：

我们考虑 $d(i, j, k)$，可以发现它等价于满足条件 $x|i, y|j, z|k, gcd(x, y) = gcd(x, z) = gcd(y, z) = 1$ 的三元组。

首先考虑两者关于每个素数独立，可以分别计算后相乘。

考虑每一个素数 p,对于 $d(i,j,k)$ 的贡献是 $(pi+pj+pk+1)$。

而对于三元组有 4 种情况：$(0,0,0),(*,0,0),(0,*,0),(0,0,*)$。

而对于三元组增加的贡献也是 $(1+pi+pj+pk)$。

那么我们就可以先固定 i，把 j 和 k 合并，公式为 $\sum_{i\le b,j\le c,gcd(i,j)=1} d[i*j]+=\lfloor\frac{b}{i}\rfloor*\lfloor\frac{c}{j}\rfloor$，通过 $b*c$ 次 gcd 来合并，gcd 可以通过 dp 快速计算，那么这步复杂度就为 $O(b*c)$。

然后把问题转化为 $\sum_{i\le a,j\le b*c,gcd(i,j)=1}\lfloor\frac{a}{i}\rfloor*d[j]$

那么这个就是经典问题了，我们把 $gcd=1$ 反演，然后把 mu 提到中间，可以使得在 $O(\sum_{i\le a}\lfloor\frac{b*c}{i}\rfloor)$ 复杂度中计算出，经过打表得到大约复杂度为 3000w。

那么我们在**时间复杂度为 $O(3000w)$，空间复杂度 $O(n^2)$** 内解决此题。

来源

CF 235E

4 The Red Button

题目大意

给出 n，现在需要构造一个长 n 的循环：a 序列。

要求满足 $a[i+1]$ 为 $a[i]*2\%\ n$ 或 $(a[i]*2+1)\%\ n$。

而且这个循环 $[0,n-1]$ 中数字只能出现一次，由于是循环所以 an 需要能转移到 a1。

数据范围 $2 \le n \le 10^5$。

题解

我们首先需要证明当 n 是奇数时是无解的。

我们先考虑当 n 是奇数时，$n-1$ 只能由 $\frac{n-1}{2}$ 得到。但 0 也只能由 $\frac{n-1}{2}$ 得到，因为 $\frac{n-1}{2}$ 只能出现一次，所以是无解的。

接下来我们考虑偶数的情况。

把 i 和 $i+1$ 配成一组，那么，可以把每条边变成两个组直接的边的话，可以发现一共有 $\frac{n}{2}$ 组，每组入度和出度都为 2，那么我们可以直接寻找欧拉环来构造方案了。

时间复杂度和空间复杂度都为 $O(n)$。

来源

CF 325E

5 Dividing Kingdom

题目大意

在二维平面上有 n 个点 (xi, yi)。

然后各有两条不重合的平行与 x 轴和平行与 y 轴的直线把平面分成 9 个部分。

每条线都不能经过点。

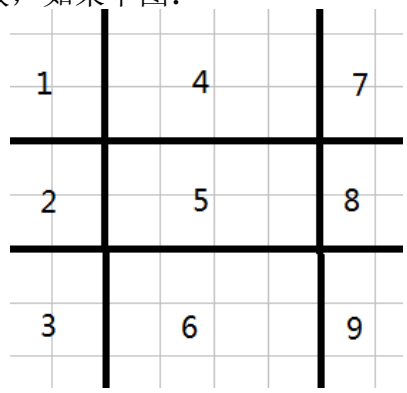
然后每个部分我们算出里面的点数。

现在告诉每个点的坐标和每个部分里的点数的 9 个数字，请你给出一组合法的直线。如果无解输出 -1。

数据范围：$1 \leq n \leq 10^5, -10^9 \leq xi, yi \leq 10^9$

题解

由于我们只知道 9 个数字，我们可以通过 9! 枚举每种每个部分对应的点数，如果下图：



然后我们便把题目转化成了每块区间中有多少个点。

这个问题我们可以通过用函数式线段树预处理算出答案。

这样此题我们可以通过在**时间复杂度在** $O(9! * 9 * log)$**，空间复杂度在** $O(n * log)$ 中解决。

来源

CF 260E

6 Princess and Her Shadow

题目大意

有两个点 (sx, sy) 和 (vx, vy)，其中 (sx, sy) 是人的位置，(vx, vy) 是影子的位置。

平面上有 n 个障碍 (x, y)，人可以向 4 个方向走，影子和人同步走，但如果影子走的下一个位子是障碍，那么影子不行动。

请构造一个行动序列使得最后人能追上影子。

数据范围：$0 \leq n \leq 400, -100 \leq x, y \leq 100$。

题解

我们可以把障碍分成 3 个状态。

1：人和影子被障碍隔开不连通。

2：人和影子被困在一个封闭区域中。

3：人和影子能到达无穷远的地方。

第一个状态是无解。

第二个状态我们可以证明只要人追着影子走必定能在地图面积长度的序列中追到。首先我们做人和影子的最短路，然后按照最短路走，然后根据影子的走的路径追。因为我们一直在追这影子走，所以每个位置人是不会走到两次的。

第三个状态我们可以先把人走到无穷远再把影子走到无穷远，然后通过最上、最下、最左、最右的 4 个障碍把人和影子重合。

最坏状态的**时间复杂度为 $O(n^3)$，空间复杂度为 $O(n^2)$**。

来源

CF 317E

7 Reclamation

题目大意

有一张 $n*m$ 的图，有 n 行 m 列，每个格子都是 4 连通，其中第 1 列和第 m 列也连通，第 0 行为源，第 $n+1$ 行为汇。

然后有 q 个操作，每个操作给一个点的坐标 (x,y)，把这个点设置为障碍。

如果这个点设置为障碍后如果源和汇不连通的话，这个操作就跳过。

问最后有几个操作能够进行。

数据范围：$1 \leq n,m \leq 3000, 1 \leq q \leq 300000$。

题解

如果没有“第 1 列和第 m 列也连通”这个条件，那么就是只需要维护一个并查集，每次询问第 1 列是否能和第 m 列 8 连通。

现在就只需要先把这张图横向复制一份，当你把一个点设置为障碍的时候，把这个点向四周 8 个点连通的时候把“环”这个条件也考虑做进去：第一列和第 $2*m$ 列连通。

那么你只需要判断 (x,y) 和 $(x,y+m)$ 是否连通就可以了。

这样**时间复杂度为 $O(q*\alpha)$，空间复杂度为 $O(n*m)$**。

来源

CF 325D

8 Levko and Sets

题目大意

给两个序列，长 n 的 a 序列和长 m 的 b 序列。

再给一个质数 p。

对于第 i 个集合是这样得到的：

1：这个集合里一开始有一个 1。

2：从集合里选一个元素 c，对于每一个 $j(1 \le j \le m)$，如果 $c*ai^{bj}\ \%\ p$ 不在集合中则加进去。

3：重复第二个操作当我们能每次加进至少一个元素到集合中。

问有多少个数在这 n 个集合中至少出现了一次。

数据范围：$1 \le n \le 10^4, 1 \le m \le 10^5, 2 \le p \le 10^9, 1 \le ai < p, 1 \le bi \le 10^9$

题解

由于 $ai < p$，所以我们可以把集合中的数表示为原根的幂次。

我们设 $ai = g^{di}$，g 为原根。

那么对于第 i 集合得到的元素就是 $di * \sum_{j=1}^{m} cj * bj\ \%\ (p-1)$，其中 cj 可以为任意非负整数。

那么可以转化公式。

设 $G = gcd(bj\ , p-1), V = \frac{p-1}{G}$

第 i 个集合得到的元素就是 $di * x\ \%\ V$。

这个很显然。

那么我们现在需要计算目前这个式子中有几个数在 n 个集合中至少出现了一次。

我们设 $hi = gcd(di, V)$。

可以发现第 i 个集合能得到的元素为 hi 的倍数。

那么，我们可以把 V 的约数都算出来，用每个集合的 hi 去更新下再算一下答案就可以了。

那么此题我们可以在**时间复杂度和空间复杂度为** $O(m)$内解决。

来源

CF 360D

9 Three Swaps

题目大意

一个长 n 的排列，一开始是：$1, 2, 3, ..., n$。

然后经过不超过 3 次的区间翻转变成了新的一个序列。

现在告诉你新的序列，要你求出一种合法的不超过 3 次的区间翻转的方案。保证数据合法。

数据范围：$2 \leq n \leq 1000$

题解

可以发现如果第 i 位有可能是翻转点的话那么需要满足 $abs(a[i] - a[i-1]) > 1$ 或 $abs(a[i] - a[i+1]) > 1$。

然后发现翻转 1 次最多产生 4 个这样的点，2 次最多 8 个，3 次最多 12 个。

那么就可以暴力搜索每次可能的翻转区间再加这个剪枝就可以了。

时间复杂度为 $O(? * n)$，空间复杂度为 $O(n)$。

来源

CF 339E

10 Ladies' Shop

题目大意

有一个长 k 的序列：$1 \leq p1 < p2 < p3, .., pk \leq m$。

现在执行一系列操作：

1：把 p 序列加进集合中。

2：每次选两个集合中的元素 x,y，如果满足 $x+y \leq m$ 而且 $x+y$ 没出现在集合中，那么把 $x+y$ 加入集合。

3：重复第二个操作直到没有新的元素加入。

现在给你这个集合中的元素，表示为长 n 的 a 数组。

让你求出一个最短长度的 p 序列。

如果无解输出 -1。

数据范围：$1 \leq n, m \leq 10^6$

题解

对于 a 序列，我们可以发现如果还能操作第二个操作的话就是无解。

最劣的的一种构造方法就是把 a 数组当成是 p 序列。

每个可能的 p 序列必定是 a 数组的子集。

然后我们通过构造可以发现如果 a 数组中的某个元素能被别的两个元素之和得到，那么这个元素必定可以不是 p 序列的，因为另两个元素也必定会得到。

那么此题就变成了把 a 数组的元素放到数轴上，然后把数轴卷积，如果有一个元素卷积后出现但 a 数组没有，那么无解。如果某个元素 a 数组出现了，但卷积后没出现，那么这个元素必须为 p 序列。

时间复杂度为 $O(m * log(m))$，空间复杂度为 $O(m)$。

来源

CF 286E

11 Theft of Blueprints

题目大意

有 n 个点的图，每条边有边权。

数据保证每 k 个点都只有一个点和这 k 个点直接相连。

这个子集的代价为这 k 个点和那个点的边权的和。

问代价的期望值。

数据范围：$2 \leq k \leq n \leq 2000, 0 \leq$ 边权 $\leq 10^9$

题解

我们可以枚举那个相连的点，算出和它相连的点数 x，对于某条和它相连的边权 y，它提供的代价为 $y * C(x-1, k-1)$。

用 double 直接做就可以了。

时间复杂度为 $O(n^2)$，空间复杂度为 $O(n^2)$。

来源

CF 332D

12 Rectangles and Square

题目大意

一个平面上有 n 个不相交的矩形,左下角为 $(x1, y1)$,右上角为 $(x2, y2)$。

问是否有一个子集的矩形能完整构成一个正方形，如果无解输出 NO。否则输出 YES 和这个子集。

数据范围：$1 \leq n \leq 10^5, 0 \leq x1 < x2 \leq 3000, 0 \leq y1 < y2 \leq 3000$

题解

首先一个合法的正方形的四角必定是某个矩形的四角。

那么我们枚举每个合法的左下角。

然后我们寻找左上角和右下角。

寻找的时候我们运用一些技巧：我们每次选择下一个可能的左上角和右下角，每次选择远的那个去验证。

由于角的个数是 $4 * n$。

然后如果是这样枚举，我们构造使得验证的次数最多，很显然的是如果把角的点分布类似正方形的状态需要验证的次数最多。

验证的话，我们可以先验证 4 个角，4 条边和内部的和，这些都是可以通过预处理算的。

那么此题的**时间复杂度为** $O(n*\sqrt{n}+3000^2)$**，空间复杂度为** $O(3000^2)$。

来源

CF 335D

13 Maxim and Increasing Subsequence

题目大意

有一个长 $n*t$ 的序列 a：$a_i = b_{((i-1)\ \%\ n)+1}$

现在给出 $n, t, maxb$ 和 b 数组且有 k 组数据，求 a 序列的最长上升子序列的长度。

数据范围：$1 \leq k \leq 10, 1 \leq n, maxb \leq 10^5, 1 \leq t \leq 10^9, n * maxb \leq 2 * 10^7$

题解

对于此题我们可以用 dp 来解决。

a 数组可以看作是 t 个 b 数组拼成，而且答案最大为 maxb。

设 $f[i][j]$ 表示为目前子序列长度为 i，最后一位在 j，目前最少是处于第 $f[i][j]$ 个 b 数组中。

转移的话，每次我们可以通过预处理算出每一位 j，后面比 bj 大的数最近在哪里，然后转移。

计算答案我们在所有 $f[i][j] \leq t$,找一个最大的 i 就是答案。

时间复杂度为 $O(k*n*maxb)$，空间复杂度为 $O(n*maxb)$。

来源

CF 261D

14 Cyclical Quest

题目大意

给一个标准串 s。

有 n 个询问，每个询问给一个串 xi，问 s 中有多少个子串和 xi 循环同构。

两个字符串被称为循环同构，如果一个串可以通过旋转获得另一个。“旋转”在这里的意思是“将字符串开头的若干字符移至字符串结尾”。例如，字符串 $abcde$ 可以被旋转到字符串 $deabc$。

字符都为小写英文。

数据范围：$|s| \leq 10^6, 1 \leq n \leq 10^5, sum(|xi|) \leq 10^6$

题解

此题是一个非常传统的后缀字符串题目。

首先可以把所有询问串复制一遍，题目就变成了对于每个询问串的每个连续长 $|xi|$ 的子串和标准串的匹配。

首先可以用后缀数组做，把标准串和询问串 $*2$ 一起做后缀数组，经典问题。

不过由于倍增做法的后缀数组自带 log 常数，使得 300w 的串长做后缀数组非常慢，无法通过此题。

我们运用后缀自动机来解决此题。

首先我们把标准串做 sam。

对于每个询问串，我们先复制一遍，从头开始从 sam 的顶端通过 sam 的儿子的数组走下去，如果没有儿子可走就走 sam 的父亲往上跳。

数组由于是匹配 $|xi|$ 的子串，当目前匹配的长度大于等于 $|xi|$，再计算答案，而且也要保证匹配长度尽量小，也是经典题目。

时间复杂度为 $O(|s|)$，空间复杂度为 $O(|s| * 26)$。

来源

CF 235C

15 Biologist

题目大意

有 n 只狗，每只狗有一个性别，可以把每只狗改变性别，需要花费代价 vi，每只狗只能改变一次性别。

有 m 个要求，每个要求规定一个性别，和选择 ki 只狗，如果这 ki 只狗的性别符合要求，那么得到收益 wi，有一些要求如果不能达到需要花费代价 g。

求最大收益。

数据范围：$1 \leq n \leq 10^4, 0 \leq m \leq 2000, 0 \leq g, vi, wi \leq 10^4, 1 \leq ki \leq 10$

题解

此题显然为网络流题目。

首先对于花费代价 g，我们可以先把这类要求花费了代价 g，然后把 wi 变成 $wi + g$。

接下来我们对于每个要求很显然可以建成一个依赖子图的形式。

对于要求之间有矛盾，把所有矛盾的要求之间建 inf 的边，由于每个要求规定一个性别，那么就可以建成二分图的形式。

这样我们得到了两张图，然后这两张图都可以运用最小割来算出答案。

但是现在我们需要把这两张图拼起来，因为第一张图中的依赖子图中有一些要求是不能共存的。

如果把第一张图的要求按照规定性别分类，我们可以发现每只狗只会和一类要求连边。

而且我们知道最小割如果把源和汇反一下是等价的。

那么我们可以对一类要求由源连入，建依赖子图，而另一类要求则和汇连入，源和狗连，由于每个狗只和一类要求连边所以不会矛盾。然后对于要求之间连第二张图的边。

那么这张图的最小割就可以保证矛盾的要求不能同时被选，且能做到依赖子图的性质。

由于此图还是二分图，所以用 dinic 跑还是有优美的时间复杂度保证。

时间复杂度为 $O((n+m)*\sqrt{m^2})$，空间复杂度为 $O(m^2)$。

来源

CF 311E

16 Polygon

题目大意

要求构造一个凸多边形，要求每个角要相同，且每条边长度不同，给定多边形点数 n。

顶点坐标的绝对值不应该超过 1000000。凸多边形的边长应该在 $[1, 1000]$ 范围内（不一定是整数）。精度在 10^{-3}，如果无解请输出 $No\ solution$。

数据范围：$3 \leq n \leq 100$

题解

当 $n \leq 4$ 时无解。

我们可以固定一条边为 1000，然后接下去来连续 $n-3$ 条边的长度都在 $[1, 2]$ 之间，长度保证能在精度内判断，最后两条边则通过直线相交计算，可以保证有极大的概率剩余两条边长度不同且不会和其他 $n-2$ 条边长度相似。

时间复杂度为 $O(n)$，空间复杂度为 $O(n)$。

来源

CF 306D

17 Endless Matrix

题目大意

有一个矩阵，里面有连续的数字，从 1 开始。如果 $a_{i,j} \leq a_{t,k}, 1 \leq i, j, t, k$ 当且仅当：

1：$max(i,j) < max(t,k)$

2：$max(i,j) = max(t,k), j < k$

3：$max(i,j) = max(t,k), j = k, j > t$

例如前 36 个数字形成的矩阵：

1	2	5	10	17	26
4	3	6	11	18	27
9	8	7	12	19	28
16	15	14	13	20	29
25	24	23	22	21	30
36	35	34	33	32	31

有 t 组数据，每组数据给定 $x1, y1, x2, y2, x1 \leq x2, y1 \leq y2$，求 $\sum_{i=x1}^{x2} \sum_{j=y1}^{y2} a_{i,j}$ 的后 10 位，如果不满 10 位，则输出答案。

数据范围：$1 \leq t \leq 10^4, 1 \leq x1, x2, y1, y2 \leq 10^9$

题解

首先我们计算矩阵的二维前缀和，那么一个区间就能变成 4 个二维前缀和的矩阵相加减得到。

对于一个二维前缀和的矩阵，我们可以把这个矩阵分成一个从 1 开始的正方形和一个多出的矩阵。

正方形里的数是连续的，可以直接计算得到。

多出的矩阵的数每行或每列也是连续的，我们只要算出每行或每列的头的和，就能直接算出答案。

以上的式子都很基础便不再细说。

接下来我们需要求出最后答案 10 位的值。

我们需要判断最后答案是否超过了 10 位。

然后可以通过模 10^{10} 直接计算答案。

对于判断是否满 10 位，我们可以通过模 3 个超过 10^{10} 级别的奇数，然后看是否模的值小于 10 位且是否相等，由于模出的数是随机的，如果 3 个数模出来都是小于 10 位且相等，那么错误的概率就可以忽略，如此就可以解决此题了。

时间复杂度为 $O(t)$，空间复杂度为 $O(1)$。

来源

CF 249E

18 Greg and Caves

题目大意

有一个 $n*m$ 的像素图。

定义一个洞为：

存在一个 $[l,r]$，使得对于 $l,l+1,..,r$ 这些行每行有两个黑点，其他行全是白点。

存在一个 t 行，使得对于满足 $l \le i \le j \le t$，第 i 行两个黑点之间的列的集合是第 j 行集合的子集。对于满足 $t \le j \le i \le r$，第 i 行两个黑点之间的列的集合是第 j 行集合的子集。

问有多少种不同的图形能表示为洞。

输出答案模 1000000007。

数据范围：$1 \le n,m \le 2000$

题解

首先我们把 $[l,r]$ 提出来相当于做为 $[1,r-l+1]$ 然后答案乘以个数。

然后对于一个区域我们由它其中第一个出现的 t 把区域分成两段，由于是第一个出现的 t 所以方案不会重复算。

那么我们先用 dp 来计算后半段的方案数。

对于每行中黑点之间的列的集合我们也只需要计算列的长度就可以了，转移的时候乘个数。

$f_{i,j}$ 表示为长度为 i，第 i 列上黑点之间的列的长度为 j 的方案数。

转移可以用前缀和优化复杂度是 $O(n*m)$ 的。

然后我们再由 f 数组来计算出 $h_{i,j}$，意义与 f 数组类似但变成整段的状态，所以当由 f 数组转移到 h数组的时候必须要保证 j 要严格下降。

那么我们就可以按照 h 数组直接来计算答案了。

时间复杂度为 $O(n*m)$，空间复杂度为 $O(n*m)$。

来源

CF 295D

19 Roadside Trees

题目大意

有一个长 n 的数轴，每个位置可以种树，每个时刻每棵树会长高 1 米。

有 m 个时刻，每个时刻你都有一个操作，操作分两种：

1：在位置 p 种一棵高度为 h 的树。

2：每次把目前位置编号第 x 小的树砍掉，每个位置如果被砍掉一棵树，那么以后也不会再种上去。

每个时刻做完操作后都要计算出目前树的最长上升子序列。

没有两棵树在某个时刻高度是相同的。

数据范围：$1 \leq n \leq 10^5, 1 \leq m \leq 2 * 10^5, 1 \leq h, x \leq 10$

题解

我们把每棵树把高度变成在某一个时刻的高度来比较。

我们把每棵树变成一个坐标 (x, y)，x 为位置，y 为高度。

然后我们以 x 为下标建一个一号线段树，以 y 为下标建一个二号线段树。

我们的最长上升子序列变成倒着的最长下降子序列。

对于加一棵树，可以发现目前比这棵树低的只可能是前 10 个时刻的树，那么我们在二号线段树中把高度比目前低的树在一号中删掉，然后在一号树中统计答案，再把那些低的树更新答案加进去。

对于删一棵树，我们把一号线段树中比目前这颗树 x 坐标小的都在二号树中删掉，然后在二号树中更新前面树的答案。

时间复杂度为 $O(m * 10 * log)$，空间复杂度为 $O(n)$。

来源

CF 264E

20 Tourists

题目大意

我们设一个直角坐标系。

某些时刻 qi 有 2 个游客同时从点 $(-1,0)$ 和 $(1,0)$ 出发，这两个游客每个时刻都移动 1 个单位，都向 y 轴正方向。

某些墙在某个时刻 ti 会在 $(0,li)$ 和 $(0,ri)$ 之间瞬间出现。

问每对游客会有多少时间看不到对方，也就是中间有墙隔着。

墙可以相交。

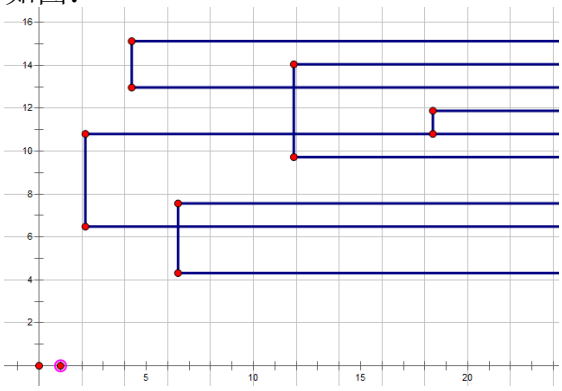
数据范围：$1 \le n,m \le 10^5, 0 \le li < ri \le 10^9, 0 \le ti,qi \le 10^9$

题解

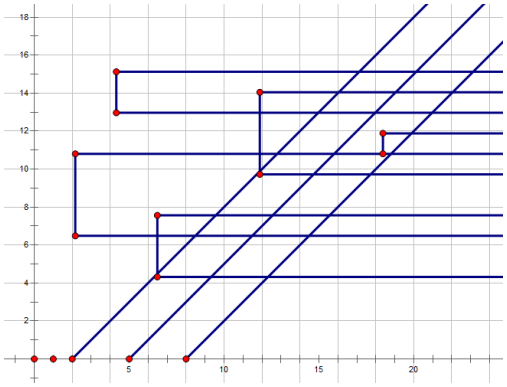
用一个坐标来表示墙的状态。

x 坐标表示时间，y 坐标表示图的 y 坐标。

如图：



斜线表示每对游客的状态：



计算每对游客有多少个时刻在墙里。

我们可以首先需要把所有的墙变成没有重叠的。

我们需要再把每条斜线转成直线，然后把墙变成头上直角三角形和矩形。

那么我们可以计算出某个平行于 y 轴上有多少墙覆盖了这条直线。

那么这个问题就是经典问题转化为扫描线就可以直接做了。

时间复杂度为 $O(n * log)$，空间复杂度为 $O(n)$。

来源

CF 286D

21 Little Elephant and Broken Sorting

题目大意

一开始有一个长度为 n 的 n 个数的排列。

有 m 次操作，每次会交换第 ai 个数和第 bi 个数，但是每个操作有一半的概率不操作。

问期望最终的序列的逆序对个数。

数据范围：$1 \leq n, m \leq 1000$

题解

对于期望的逆序对个数，我们可以算出每对 (i, j) 的逆序概率，答案为总和。

那么我们只需要用 $f[i][j][k]$ 表示在第 i 个操作前位置为 j 和 k，在最后时刻 j 的位置比 k 的位置前的概率。

转移 $O(1)$，但由于每次交换只交换两个数，那么每次交换只会改变 $O(n)$ 的状态。

那么**时间复杂度为 $O(n * m)$，空间复杂度为 $O(n * m)$**。

来源

CF 258D

22 Dima and Figure

题目大意

有一张 $n*m$ 的图，初始每个格子都是白色。

现在要在其中作画，把一些格子涂成黑色，要求：

包含至少一个黑色，黑色的格子形成一个四连通块，且要求从任意一个黑色的格子 $(x1,y1)$ 移动到另一个任意黑色的格子 $(x1,y1)$ 的最短路长为 $|x1-x2|+|y1-y2|$。

数据范围：$1 \leq n,m \leq 150$

题解

可以发现从要求中发现涂的形状必须为凸包。

那么我们可以运用 dp 解决。

用 $f[i][j][k][u][r]$ 表示做到第 i 行，黑色的列为 $[j,k]$，u 和 r 表示左边和右边是否开始下降。

转移的时候运用前缀和优化转移。

转移的时候注意要先缩小再扩大，不然可能会导致图不连通。

时间复杂度为 $O(n*m*m*4)$，空间复杂度为 $O(n*m*m*4)$。

来源

CF 273D

23 Cow Tennis Tournament

题目大意

有 n 个人，每个人有一个能力值 si，每个人能力值不同。

每两个人之间都要进行比赛，每次比赛能力值大的获胜。

现在开始有 m 次操作，对于一次操作，如果两个人的能力值都在 $[xi, yi]$ 中，那么这两个人比赛的成绩就变反，每场比赛的结果可以多次改变。

问有多少个三元组 (p, q, r) 使得 p 打败 q，q 打败 r，r 打败 p。

数据范围：$3 \leq n, m \leq 10^5, 1 \leq si \leq 10^9, 1 \leq xi < yi \leq 10^9$

题解

合法的答案为全集减去不合法的。

我们现在需要计算不合法的答案。

我们可以发现合法的情况是每个人胜一次和败一次，不合法的每种情况都必定有一个人败了两次，那么我们只需要计算每个人被几个人打败，那么答案贡献就是在打败他的人数中的二元组的个数。

然后我们把人按照 s 排序，需要计算 s 比其小的但被奇数个操作覆盖和比其大的但被偶数个操作覆盖的个数，这个我们可以用线段树来维护。

时间复杂度为 $O(n * log)$，空间复杂度为 $O(n * log)$。

来源

CF 283E

24 Olya and Graph

题目大意

有 n 个点，m 条有向边的图，问现在有多少种添边方案能使得满足一下要求：

1：从 i 出发能到达 $i+1, i+2, ..., n$。

2：每条边从 u 到 v 要满足 $u < v$。

3：两点之间最多一条边。

4：对于一对点 $i < j$，如果 $j - i \leq k$，那么两点间有 $j - i$ 条边。

5：对于一对点 $i < j$，如果 $j - i > k$，那么两点间有 $j - i$ 条边或 $j - i - k$ 条边。

方案数模 1000000007。

数据范围：$2 \leq n \leq 10^6, 0 \leq m \leq 10^5, 1 \leq k \leq 10^6$

题解

由要求可知，如果需要有解，那么必须使得边要不是 $i \to i+1$ 就是 $i \to i+k$。而且需要连的边如果跨越了长度 k，那么这些边必须相交。

那么我们先对读入的边判断是否无解。

然后我们可以枚举跨越了长度 k 的边的最后一条边的位置，那么前面可以用二的幂次来计算方案数就可以了。

时间复杂度为 $O(n)$，空间复杂度为 $O(n)$。

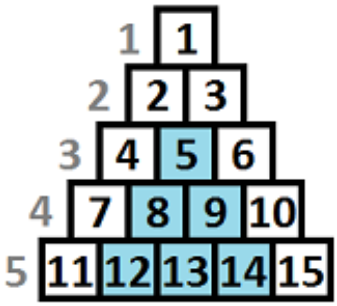
来源

CF 305D

25 Transferring Pyramid

题目大意

有一个 n 层金字塔如图形状：



上图为 $n = 5$ 的情况，每个数字表示一个位置。

现在有两种操作：

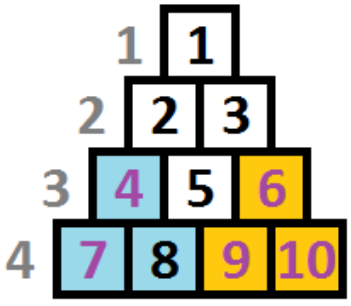
1：为 $t\ i\ v$，其中 $t = 1$，表示把 i 这个位置上值改成 v。

2：为 $t\ i\ v1\ v2...vs$，其中 $t = 2$，表示把 i 这个点为顶点向下的小金字塔的值按位置下标位置按顺序改为 $v1\ v2...vs$。

如下图为 $n = 5$，且操作为

2 4 $v4$ $v7$ $v8$

2 6 $v6$ $v9$ $v10$ 的情况：



我们定义坐标 (x,y)，为第 x 行从左向右第 y 个。

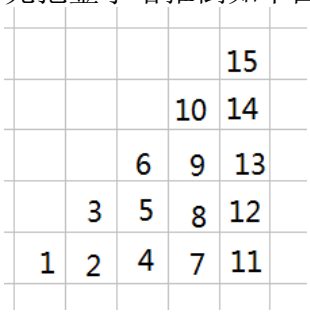
我现在给出 n 和 m 个坐标，要求 m 个坐标至少被赋值过一次，求操作的数字最少的个数。

数据范围：$1 \leq n, m \leq 10^5$

题解

我们可以发现现在有两种方案，一种是花 3 个数字把一个坐标标记，另一种是花 $2 + sum$ 把以一个点为顶的 sum 个坐标标记。

先把金字塔推倒如下图：



我们可以发现如果是选 $t = 2$ 那种操作一定不会在太上层因为需要赋值的坐标只是 10^5 的级别，那么赋值的不会超过底部向上 1000 层的。

那么我们就可以把金字塔推倒后一层一层向上做。

我们用 $f_{i,j}$ 来表示到第 i 层最后 j 列已经被第二种操作覆盖了。

然后当从第 i 层向第 $i + 1$ 层转移的时候，每一列都要向后推一列，然后再枚举此层中最前的那列用第二个操作转移。

那么此题**时间复杂度为** $O(n * \sqrt(n))$，**空间复杂度为** $O(n * \sqrt(n))$。

来源

CF 354D

26 Maxim and Calculator

题目大意

有一个计算器，有两个值 a, b。

一开始 $a = 1, b = 0$，有两种操作：

1：$b \to b + 1$

2：$a \to a * b$

求有多少个数 $x(l \le x \le r)$ 满足能经过不超过 p 次操作得到。

数据范围：$2 \le l \le r \le 10^9, 1 \le p \le 100$

题解

可以发现得到的数的所有质因数是不会超过 p 的。

然后经过打表可得在不超过 10^9 的数中质因数不超过 100 的只有不超过 300w 个。

那么我们可以直接 dp 计算每个数字最少多少次操作得到。

hash 一个数可以直接用递推计算。

时间复杂度为 $O(300w * p)$，空间复杂度为 $O(300w * p)$

来源

CF 261E

27 Printer

题目大意

有一个打印机，从时刻 0 开始工作，每秒打印一页纸。

某些时刻会有一些任务，接到的时间为 ti，需要打印 si，pi 表示优先级。

每个时刻我们有一个队列表示目前需要打印的任务，每次我们都会选择目前队列任务中优先级最高的一项任务，然后打印一页。

但是对于第 x 个任务，我们并不知道 px，但我们知道这个任务打印完最后一页的时刻 T。

求一个可行的 px，注意 p 互不相同。

数据范围：$1 \leq n \leq 50000, 0 \leq ti \leq 10^9, 1 \leq si, pi \leq 10^9$

题解

我们把其他的 p 排序，相邻两个 p 直接的优先级对情况影响是一样的，那么我们枚举每个不同情况的 px。

我们根据优先级从小到大枚举任务，对于每个时刻如果加入 x 任务结束时刻等于 T 的话为解。

计算结束时间我们用线段树来操作。

时间复杂度为 $O(n * log)$，空间复杂度为 $O(n * log)$。

来源

CF 253E

28 Sheep

题目大意

每个羊控制一个区域 $[li, ri]$。

对于某个位置，把控制这个位置的所有的羊绑在一起。

现在把羊排在一列使得绑在一起的相差最远的距离最近。

求构造这个方案。

数据范围：$1 \leq n \leq 2000, 1 \leq li \leq ri \leq 10^9$

题解

对于这个距离我们可以二分答案 ans。

对于每只羊，我们按照 li 升序排序。

一开始每只羊最劣的可能位置都是 n。

我们定义这个值为 lasti，再定义 tj 表示为 $lasti \leq j$ 的个数。

然后我们逐位确定羊的顺序。

对于每一位我们都选择目前有可能成为答案中的 ri 最小的一个为目前位置的羊。

如果一个值 j 不可能成为答案则存在一个 k 使得 $k < lastj \, and \, k = tk$。

而如果无解的话则是存在一个 k 使得 $tk > k$。

然后我们确定了目前这位之后我们再更新剩余值的 last 数组。

那么此题**时间复杂度为 $O(n^2 * log)$，空间复杂度为 $O(n)$**。

来源

CF 309E

29 Summer Homework

题目大意

有一个序列 $a1, a2, ..., an$，有 m 次操作，有 3 种：

1：把 ax 赋值成 v。

2：询问 $\sum_{x=0}^{ri-li} f_x * a_{li+x}, f_0 = f_1 = 1, f_i = f_{i-1} + f_{i-2}$。

3：把 al 到 ar 加上 d。

对答案模 1000000000。

数据范围：$1 \leq n, m \leq 200000$

题解

对于乘 f，我们可以把 f 拆成矩阵乘法的形式。

然后可以把一个矩阵看成一个数，而且乘的系数是不变的。

那么此题就变成了经典问题，用线段树直接可做。

时间复杂度为 $O(n * log * 8)$，空间复杂度为 $O(n * 8)$

来源

CF 316E

30 Colorado Potato Beetle

题目大意

有一张 $(10^{10}+1)*(10^{10}+1)$ 的地图，现在人站在地图中间。

人移动 n 次，每次向上下左右四个方向移动一段距离 xi,并在经过的格子上标记。

现在问有多少点无法和地图外部四连通。

数据范围：$1 \leq n \leq 1000, 1 \leq xi \leq 10^6$

题解

把地图转折点坐标离散，直接做 bfs 就可以了。

注意离散的时候如果中间有点，中间也要加进去一个点。

时间复杂度为 $O(n^2)$，空间复杂度为 $O(n^2)$。

来源

CF 243C

31 Good Substrings

题目大意

给一个标准串 s，再给 n 个要求串 pi。

问 s 中有多少个不同的子串满足每个要求串的要求。

如果满足第 i 个要求的话，需要满足这个子串在 pi 中出现的次数为 $[li, ri]$。

字符都为小写英文。

数据范围：$1 \leq n \leq 10, 1 \leq |s|, |pi| \leq 50000$

题解

此题用各种后缀数据结构都可以解决，这里介绍一个用后缀数组的方法。

把标准串和每个要求串放在一起做后缀数组。

对于每一个标准串的后缀，我们可以运用倍增求出当这个子串在要求串出现的次数符合要求的时候长度的限制。这个可以先求出 height 数组，然后在在 height 数组上做 倍增预处理，然后对于每个后缀二分可行的长度，用倍增验证。

时间复杂度为 $O(n * |s| * log^2)$，空间复杂度为 $O(n * |p|)$。

运用 sam 直接 dfs，可以做到**时间复杂度为 $O(n * |s|)$，空间复杂度为 $O(n * |p| * 26)$。**

来源

CF 316G

32 Tree and Table

题目大意

有一颗 $2*n$ 个节点的树，现在要把这棵树放进 2 行 n 列的表格中，有一下要求：

1：表格的每个格子放入一个节点。

2. 如果两节点间有边相连，那么它们放入的格子是相邻。

求把树放进表格的方法数。两种方法不同当且仅当存在一个节点放入的格子不同。

答案模 1000000007。

数据范围：$1 \leq n \leq 100000$

题解

如果 $n = 1$，答案是 2。

如果有一个点度数 > 3，答案是 0。

如果没有度数为 3 的点，答案是 $2 * n^2 - 2 * n + 4$。

证明：

认为 1 号点度数为 1。

如果 1 号在边界上，方案数为 n，否则由于一边固定方案所以是 $n - 1 \leftrightharpoons (i - 1 + (n - i))$。

头尾反一下答案还要 $*2$

方案数可以用归纳法证，把 $n \to n + 1$，那么当两个度数为 1 的点在同侧边界上，那么答案有两种，否则答案只有 1 种。

所以 $n \to n + 1$，方案数 $+1$。

当有度数为 3 的点时：

1) 先选一个点当根。

然后枚举他的 3 个儿子的顺序，一个左一个右一个下。

然后下的那个有儿子的再枚举。

现在变成了子问题。

a) 一个矩阵，左上有点的方案数。

b) 一个矩阵，左上和左下有点的方案数。

现在把 b) 的问题转化到 a) 上去。

就是判一些儿子的情况，因为一定有一支的儿子先走完。

然后就是做 a) 了。

那么又有两种情况。

如果那个点 x 度数是 2 或者是 3。

如果是 3 的话，我们可以把这两个儿子的接下去的度数是 2 的点都并掉，那么就变成了 a) 了。

现在处理度数是 2 的点。

如果他的子树没有度数是 3 的点了。那么答案就是有一坨点，起点在边界上，求链的方案数。

如果他的子树有度数是 3 的点。

我们找到最近的度数为 3 的点，设为 w。（w 上面就是链）

那么对于 w，有两种可能。

a.1) 从左边过来

a.2) 从列上过来

设 w 的两个儿子为 x, y

对于 a.2)，如果 x, y 的子树都没有度数为 3 的点，显然 $O(1)$。

如果 x, y 的子树都有度数为 3 的点,那么答案为 0。

所以只能是有度数为 3 的点放右边。

那么复杂度还是 $O(1)$，递归 a)。

对于 a.1)，和 a.2) 较相似，但子树中有度数为 3 的点能放右边和同一列上，但是递归的时候可以发现，如果运用记忆化搜的话，相邻的两个度数为 3 的点之间的路径只会被遍历到两次。

那么复杂度就正确了。

这道题就是各种细节考虑。

时间复杂度为 $O(n*$ 常数$)$，空间复杂度为 $O(n)$。

来源

CF 251E

33 Liars and Serge

题目大意

有 n 个人，每个人说谎话或说实话，每个人都知道别人的属性。

现在问每个人有多少人讲实话，讲实话的人说出准确的数字，说谎话的人会说出 $[1,n]$ 的一个数字但不是正确答案，每个说谎话的人都不会考虑别人的回答。

现在给出每个人的答案，问有 k 人显然是说谎。

求不同的每个人答案的方案数，只要两个方案有一个人的回答不一样那么就算不同。

答案模 777777777。

数据范围：$1 \le k \le n \le 2^8$，n 是 2 的幂次。

题解

如果一个人显然说谎，设他说的数字是 ai，那么 ai 这个数字出现的次数是不等于 ai 的。

那么我们可以运用 dp 来解决此题。

$f[i][j][k]$ 表示目前到了第 i 个数字，目前有 j 个人说谎，一共用了 k 个人，然后我们枚举说出 $i+1$ 这个数字的人数，然后用组合数转移。

这个**时间复杂度为 $O(n^4)$，空间复杂度为 $O(n^3)$**。

但由于 n 是 2 的幂次，状态数只有 2^{11} 个，那么我们可以运用打表直接解决。

时间复杂度为 $O(1)$，空间复杂度为 $O(n*8)$。

来源

CF 256D

34 GCD Table

题目大意

有一个 n 行 m 列的表格 G，$G_{i,j} = gcd(i, j)$。

现在给你一个长度为 k 的 a 序列，问你这个序列是否在表格某行连续出现。

数据范围：$1 \leq n, m, ai \leq 10^{12}, 1 \leq k \leq 10000$

题解

设 a 序列匹配的第一位是第 x 行第 y 列。

我们首先可以发现 $lcm|x$，而且如果 x 变大符合条件的概率就要低，因为 x 里会多了别的质因数。所以 $x = lcm$。

然后我们可以发现 $y\%\ a1 = 0, (y+1)\%\ a2 = 0, ...$，就相当于做一个模数不互质的中国剩余定理。

我们计算出一个最小的合法的 $y = k$，由于合法的 y 为 $k + lcm*?$，那么我们很显然只要验证 (lcm, k) 就可以了。

做模数不互质的中国剩余定理时我们只需要把每个式子拆开成每个模数为质因数或质因数的幂次再拼起来就可以了。

时间复杂度为 $O(k * log^2)$，空间复杂度为 $O(k)$。

来源

CF 338D

35 More Queries to Array...

题目大意

有一个长 n 的 a 数组，有 m 个两种操作：

1：把 a 数组的 $[l,r]$ 的值赋成 x。

2：计算 $\sum_{i=l}^{r} a_i * (i-l+1)^k, 0 \leq k \leq 5$。

答案模 1000000007。

数据范围：$1 \leq n \leq 100000$

题解

经典线段树数据结构问题。

时间复杂度为 $O(n*log*5*5)$，空间复杂度为 $O(n*5)$。

来源

CF 266E

36 String Theory

题目大意

一架竖琴有 n 行 m 列构成的 $n*m$ 的网格。

左右两侧每侧有 n 个钉子，上下每侧有 m 个钉子。

竖琴有 $n+m$ 根琴弦，每根琴弦两端分别固定在不同侧的边界上。

每个钉子上有且仅有一根琴弦。

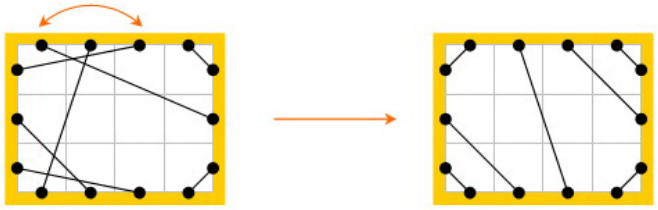
如果存在两根相互交叉的琴弦，那么竖琴将无法弹奏。

为了能够正常弹奏有以下两种操作：

1：选择不同的两列，对应地交换处于同侧的钉子（两侧必须同时交换），但不改变钉子与其所固定的琴弦的连接。

2：选择不同的两行，对应地交换处于同侧的钉子（两侧必须同时交换），但不改变钉子与其所固定的琴弦的连接。

如图为交换两列的情况：



求出对于初始时的每行每列，在最后的竖琴上所应该处在的位置或无解输出 $No\ \ solution$

数据范围：$1 \leq n, m \leq 10^5$

题解

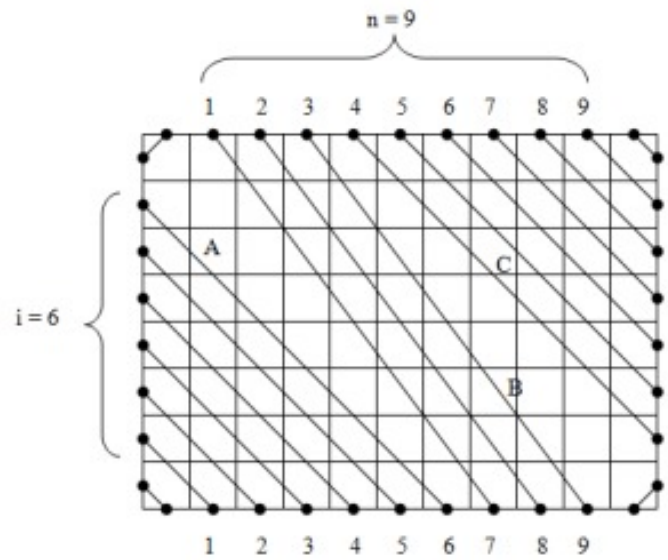
我们把琴弦分成 6 类：左上，左下，右上，右下，上下，左右。

首先可以发现对于任意操作，这 6 类琴弦的数量不变。

如果同时拥有上下和左右的琴弦那么无解。

否则我们可以通过选择默认没有左右类琴弦。

然后我们思考最终的情况可以发现，结果必定是如下图所示：



最终结果构了很多的环。

如果前 4 类弦有的话就把前 4 类弦放到角上去。

然后把上下的边放中间，此外要不是左下和右上要不就是左上和右下。

然后把琴弦按照图中摆放整齐。

我们可以发现一些性质：除了角上由 4 条不同的琴弦构成的环外多出来的琴弦构成的环本质都是相同的！只是位移不同而已。

我们把角上的琴弦删去之后，我们可以发现得到的就是一个矩阵，如上图宽为 i 长为 n，然后每条琴弦长的位移都是 i，那么我们可以得到环的个数就是 $t = gcd(i, n)$，每个环的弦的条数为 $\frac{i+n}{t}$。

然后我们把环的标准的每条边的类型按顺序做成一个序列。

再对于原图，我们找环，如果环是可以放到角上的，我们把它放到角上，否则我们也把环的序列搜出来和标准串做 kmp，如果不能完全匹配或者环数不够就是无解，否则就把一个一个环填到最终的图形中。

由此我们就可以求出原来的每行和每列对应最终的排列了。

时间复杂度为 $O(n + m)$，空间复杂度为 $O(n + m)$。

来源

CF 269E

37 Binary Key

题目大意

给一个标准的串 p，要求构造一个字典序最小的 01 串 q，使得经过下列伪代码后能得到要求串 s。

```
i = 0;
j = 0;
s = <>;
while i is less than the length of the string p
{
    if q[j] == 1, then add to the right of string s character p[i];
    increase variables i, j by one;
    if the value of the variable j equals the length of the string q, then j = 0;
}
```

数据范围：$1 \leq |p| \leq 10^6, 1 \leq |s| \leq 200, 1 \leq k \leq 2000$

题解

我们可以枚举 q 序列中有多少个 1。

然后我们可以知道每一位为 1 得到的序列是什么，我们还需要知道 q 序列第几个 1 需要得到的序列是什么。

我们倒着枚举就可以算出固定 q 序列 1 的个数中计算出合法的字典序最小的串。

那么我们把所有可行的串取一个最优串就可以了。

时间复杂度为 $O(|s| * |s| * k)$，空间复杂度为 $O(|p|)$。

来源

CF 332E

38 Deja Vu

题目大意

有 n 个点 m 条有向边的图，两个点最多一条边而且没有自环。

每经过一条边都能看到一些点的编号，可以经过一条边多次。

如果一条路径为好的路径则要求路径上的点构成的序列和路径上边编号构成的序列完全一样。

要求统计长度为 $1, 2, .., 2 * n$ 的不同长度的好的路径方案数。

答案模 1000000007。

数据范围：$1 \leq n \leq 50, 1 \leq m \leq \frac{n*(n-1)}{2}$, 边上写的数字的总个数 $\leq 10^5$

题解

此题要分成两个部分做。

每条好的路径都可以由 3 个部分拼起来：

1：一个最短的好的路径。

2：一条路径尾端的点不在边序列中。

3：一条路径头部的点不在边序列中。

a)：然后两条好的路径的一个尾和一个头如果之间的边没有任何点的编号，那么可以拼成一条长的路径。

然后对于某条最短的好的路径，我们可以发现一定有一条边的点的序号包含了这两个点，然后头部按照目前的序列搜索直到点序列和看到的序列的头符合或过长判无解。尾部也同样操作。如此我们可以得到最短的好的序列。

而对于 2 和 3 我们也可以用相似的做法得到。

然后我们用 dp 计算方案数。

首先我们可以发现 $2 + 1$ 和 $1 + 3$ 都能得到新的 1 号路径。

那么我们可以用 $f[i][j][k]$ 表示路径长度为 i，头为 j，尾为 k，的方案数。

为了防止答案重复计算，我们需要把 1 全部做完再做 2。

因为 2 和 3 只有 $O(n^2)$ 条，那么我们可以在 $O(n^4)$ 的复杂度内计算出答案，然后我们通过运用 a) 方案计算最终的答案。

为了防止答案重复计算，我们使得每次把两条路径合并的时候只能用 f 合并在新的路径的尾部。

那么我们重新定义一个数组 $h[i][j]$ 表示长度为 i，尾部为 j，的方案数。

那么我们也可以在 $O(n^4)$ 的复杂度内计算出答案。

所以**时间复杂度为 $O(n^4)$，空间复杂度为 $O(n^3)$**。

来源

CF 331E

39 Xenia and String Problem

题目大意

有一个字符串 s。

如果 s 满足以下条件，那么称为是好的串：

设 mid 为 $\frac{|s|+1}{2}$。

1：$|s|$ 是奇数。

2：$s[mid]$ 在 s 中只出现一次。

3：$|s| = 1$ 或 $s[1..mid-1]$ 和 $s[mid+1,..,|s|]$ 相同且都是好的串。

定义 p 的价值为 p 的所有子串如果是好的串那么贡献为串长的平方。

现在给出串 t，最多允许 t 中修改一个字符，使得串的价值最大。

字符都为小写英文。

数据范围：$1 \leq |t| \leq 100000$

题解

可以发现好的串长只有 log 种，所以总共只有 $nlogn$ 个串。

如果不改变直接计算答案，我们可以枚举每个可能的子串用 hash 判断。

我们可以发现每个串如果可能成为解那么要不是中心要不就是两端只有一位不同，那么变成好的串的可能修改的次数就是 $nlog * 26$ 级别的。

那么我们就可以想出我们的做法了。

我们枚举每一位和每一位修改后的值，我们开始递归，如果目前这个串是好的串我们尝试向左边和右边翻一倍判断是否相等，我们把 hash 值一起递归进去能更优美地实现代码。

这样我们搜索的复杂度就等于变成好的串的个数了，复杂度我们上文已证明。

那么我们的此题**时间复杂度为 $O(nlog * 26)$，空间复杂度为 $O(n)$**。

来源

CF 356E

40 Dima and Game

题目大意

有两个人在玩博弈游戏。

现在写下 n 对数字 $(l[i], r[i])$，玩家轮流操作，操作如下：

选择一对数，满足 $l[i] + 2 < r[i]$，设 $x = \lfloor \frac{r[i]-l[i]}{3} \rfloor$，然后把这对数替换成 $(l[i] + x, l[i] + 2 * x)$ 或 $(l[i], r[i] - x)$。

当不能操作算输。

问有多少种情况的 n 对数字能使得后手胜。

如果这些数对被按照不同的顺序写了下来，则将其视作不同的方法。

答案模 1000000007。

数据范围：$1 \leq n \leq 1000, 1 \leq li < ri \leq p \leq 10^9$

题解

sg 问题。

首先每对数字互不相关，那么可以把每对的 sg 算出来异或得到全局的 sg。

对于每对的 sg，我们发现每对情况只和 $r[i] - l[i]$ 有关。

那么我们可以直接递推 sg。

由于值有 10^9 大，我们可以运用打表。

打表后把相同的合并可以发现只有 100 多个数字了。

由于转移只有两种，所以答案 sg 值只会是 0/1/2。

我们预处理出每一种 sg 值可能的数量。

我们要使得后手必胜，那么 n 个 sg 异或要为 0。

直接暴力或 dp 计算。

时间复杂度为 $O(n * 100)$，空间复杂度为 $O(n * 100)$。

来源

CF 273E

41 Lucky Tickets

题目大意

一个数字序列如果被称为 $k-lucky$ 要求把这个数字序列每一位拆开，在一些数的左边或右边加上加减乘或括号使得最后答案等于 k。

现在给出 k，求出 m 个不同的长 8 的数字序列。保证有解。

数据范围：$0 \leq k \leq 10^4, 1 \leq m \leq 3 * 10^5$

题解

很厉害的构造题。

首先由于有 8 位而且 k 比较小。

那么我们可以固定 4 位使得答案为 k。

那么我们要让前 4 位方案尽可能多。

那么我们可以枚举前 4 位的数字，然后把位数拆开，在中间加运算符，如果值小于 10000，那么再由后 4 位固定，然后前 4 位和后 4 位也可以互相交换，那么可以满足 30w 个的要求。

用 set 判重。

时间复杂度为 $O(m * log)$，空间复杂度为 $O(m)$。

来源

CF 333C

42 TorCoder

题目大意

有一个长 n 的字符串，有 m 个操作。

每个操作选择一个区间 $[l,r]$，如果把 $[l,r]$ 中的字符重排能使得其变成字典序最小的回文串，那么修改，不然跳过这个操作。

问做完 m 次操作后的字符串。

字符集是小写英文。

数据范围：$1 \leq n,m \leq 10^5$

题解

这个可以运用线段树来解决。

我们把每个字符分开，分成 26 个线段树。

我的线段树需要能计算区间中的每个字符有几个，最多只有一种字符个数是奇数才合法。

然后由于字典序最小的回文串，那么方案固定，而且回文串最多会分为近 50 段相同的子串。

我们先把线段树清空。

再对每一段线段树覆盖就可以了。

时间复杂度为 $O(m*log*26)$，空间复杂度为 $O(n*26)$。

来源

CF 240F

43 Playing with String

题目大意

一开始有一个串 p，两个人玩博弈问题，轮流操作。

每次操作可以选择一个串 s，然后选择一个位置 i，满足 $1 < i < |s|, s[i-1] = s[i+1]$。

然后把这个串分成：$s[1..i-1], s[i], s[i+1..|s|]$ 三个串。

问必胜的先手或是后手。

如果先手必胜，那么输出第一步操作，如果有多个解输出最小的 i。

数据范围：$1 \leq |s| \leq 5000$

题解

sg 问题。

这个串可以看成是 01 串。

1 表示这位左右的字符相同。

然后我们每次选一个 1，然后把串分开。

所以可以发现一些连续的 1 之间是互不影响的。

那么我们可以算出有连续个 1 个 sg，然后异或得到答案。

那么我们可以用递推计算 sg。

对于方案，我们可以枚举每一位，然后计算 sg 判断结果。

时间复杂度为 $O(n^2)$，空间复杂度为 $O(n)$。

来源

CF 305E

44 Graph Game

题目大意

现在有 n 个点 n 条边连通的图。

现在我们递归做。

对于一个连通图，我们随机一个点，然后把这个点删掉，把图分成很多个连通块，对于每个连通块我们都递归做。

设 sum 为每次递归的时候连通块的点数和。

问期望 sum 是多少。

数据范围：$3 \leq n \leq 3000$

题解

我们计算每个 x 为选中的点的时候，这个连通块的点数个数的期望值。

当是一棵树的时候。

当以 x 为选中的点的时候，对于 y，设 v 为 x 和 y 之间的点数个数。

那么要使得 y 在这个连通块，我们把选中的点按顺序列成一个序列，这样每个序列能表示成一种方案，那么就要求这 v 个点在序列中 x 是最前的。

那么 y 在这个连通块的概率就是 $\frac{1}{v}$。

现在不是树而是环套外向树。

那么有些对点之间不是一条链而是一条链加一条边。

我们就计算这对点之间的点的某条路径中处在环上点有几个。

我们设某条路径总长为 y，处在环上的点有 z 个，环长为 x。

那么我们就可以计算概率了。

如果 $z \leq 1$，那么概率为 $\frac{1}{y}$。

否则是在环上，那么我设 $a = y - z + x$，表示和这两个点之间有关的点数。

概率 = 只需要在 $a-(z-2)$ 个中固定一个点为最前的概率 + 只需要在 $a-(x-z)$ 个中固定一个点为最前的概率 - 要在所有点中固定一个点为最前的概率。

所以**时间复杂度为 $O(n^2)$，空间复杂度为 $O(n)$**。

来源

CF 235D

45 Xenia and Dominoes

题目大意

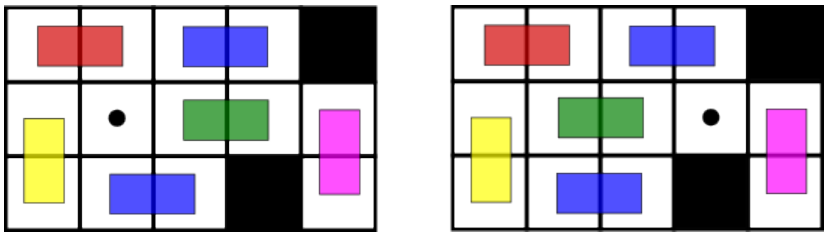
有一个 $3*n$ 的矩阵，有些格子是禁止的，还有一个格子是要求的。

现在要把剩余的格子里用 $1*2$ 的多米诺骨牌覆盖。

每个剩余的格子都必须被一个多米诺骨牌覆盖。

然后还要求这个方案中被要求的格子能至少移动一次。

移动就是把一个多米诺骨牌移到被要求的格子中，横向的多米诺骨牌只能横向移动，纵向的多米诺骨牌只能纵向移动。你不能旋转多米诺骨牌。下图表示一个合法的移动：



求不同的方案数。

两个拼图被认为是不同的当存在一对格子在一个拼图中有一个多米诺骨牌而在另一个中没有。

答案模 1000000007。

数据范围：$3 \leq n \leq 10^4$

题解

由于只有 3 行，那么怎么做都可以了。

我运用了插头 dp，不过，容斥移动的状态加位运算 dp 也可以做。

时间复杂度为 $O(n*3*2^4)$，空间复杂度为 $O(1)$。

来源

CF 342D

46 Ping-Pong

题目大意

每个时刻都有一个区间的集合，每次可以从区间 (a,b) 移动到区间 (c,d) 的限制是要满足 $c<a<d$ 或 $c<b<d$。

有 m 个操作，有两种：

1：可以加入一个区间 (a,b)，满足当前加入的区间长度严格大于以前加入的区间的长度。

2：询问第 x 个加入的区间是否能够移动到第 y 个加入的区间。

数据范围：$1 \leq n \leq 10^5$

题解

由于有“当前加入的区间长度严格大于以前加入的区间的长度”这一条件，所以对于现在加入的区间只能是和以前的区间互相到达或以前的能到达目前这个。

我们现在把可以互相到达的用并查集缩起来。

那么我们就可以计算对于现在这个区间，我们根据端点处可以知道有哪些区间包含端点这个位置，那么说明目前这个能到达以前的那个，那么我们可以把这两个缩起来，因为能互相到达。

对于询问时，我们只需要判断是否在一个集合里或第 x 个的区间能直接到第 y 个的集合就可以了。

注意，不能是第 x 个的集合能直接到第 y 个的集合，因为如果两个集合相等就不一定了。

然后我们可以用线段树来实现我们的思路。

我们把区间里加一个目前现在这个编号。

然后对于端点处，我们在线段树包含这个端点里的编号全合并，然后删掉，只加入目前这个编号。

那么**时间复杂度为** $O(n*log*\alpha)$，**空间复杂度为** $O(n*log)$。

来源

CF 319E

47 Yaroslav and Algorithm

题目大意

有一个算法。

1：输入一个字符串为 a。

2：算法由一些命令组成。i 号命令形式为 $si >> wi$ 或 $si <> wi$，其中 si 和 wi 是长度不超过 7 的字符串（可以为空），由数字或？组成。

3：这个算法每次寻找一个编号最小的命令 i，使得 si 是 a 的子串。如果没有找到，那么整个算法终止。

4：设找到的命令编号为 k。在字符串 a 中，sk 第一次出现的位置会被 wk 替换。如果这个命令形如 $sk >> wk$，那么这个算法回到第 3 步。否则，算法终止。

5：算法的输出就是算法终止时字符串 a 的值。

现在要输入一个数，使得输出这个数加一。

要求构造这个算法，有以下要求：

1：命令的条数不能超过 50。

2：算法必须对于每个输入都执行不超过 200 步。

有 t 组数据。

数据范围：$1 \leq$ 数 $\leq 10^{25}, 1 \leq t \leq 100$

题解

构造题。

我们运用？来解决此题。

那么我们先在头上加一个?。

然后把？移到末尾。

把？变成??。

然后开始把末尾加一。

如果进位了，也处理一下，把?? 放到前面去。

然后就完成了整个算法。

我们把进位放在第一条。

那么只有 30 多个命令，每个输入做的次数是输入的数字长度级别的。

时间复杂度为 $O(1)$，空间复杂度为 $O(1)$。

来源

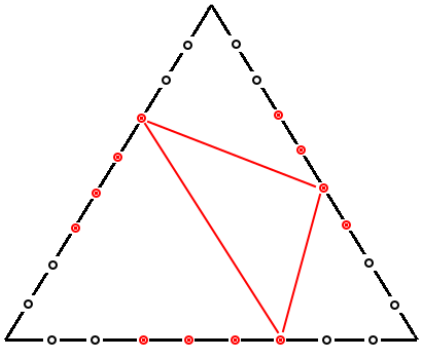
CF 301C

48 Tennis Rackets

题目大意

有一个正三角形的球拍。

每条边都有 n 个洞，但是靠近每个角的 m 个洞都是不能用的，如果下图：



现在要每条边中选一个空，然后相互连边使得构成一个三角形。

问有多少种可能的构成的钝角三角形。

两个三角形算不同，当且仅当它们固定在某一位置拍照片是不同的。（即不考虑旋转同构）

数据范围：$1 \leq n \leq 32000, 0 \leq m \leq \lfloor \frac{n}{2} \rfloor$

题解

暴力就可过了。

我们把左下角设原点，建立坐标系，算出每条边的直线解析式。

那么对于某个三角形，我们就可以算出每个点的坐标。

我们确定两个点，然后我们再确定一个点这个点的角是钝角，那么我可以运用点积来确定出另一个点的范围。

那么，我们可以在 $O(n^2)$ 的复杂度算出答案。

由于固定了钝角的角，那么我们把答案乘 3 就是正确答案。

时间复杂度为 $O(n^2)$，空间复杂度为 $O(n)$。

来源

CF 309D

49 Yaroslav and Arrangements

题目大意

当一个数列 a 是优美的话，要求如下：

$r = |a|$

1：$|a1 - a2| = 1, |a2 - a3| = 1, .., |ar - a1| = 1$。

2：$a1 = min(a)$。

当数列 b 是优秀的话，要求如下：

1：$r = |b|, 1 \le r \le n, 1 \le b \le m$

2：$b_i \le b_{i+1}$

3：通过重排数列中的元素可以得到至少一个至多 k 个不同的优美的数列。

现在给出 n, m, k，求不同的优秀的数列的个数。

答案模 1000000007。

数据范围：$1 \le n, m, k \le 100$

题解

此题有各种 dp 的姿势。我这里讲述一种。

首先可以发现优美的数列是呈山峰形状的。

然后我们可以发现我们只需要控制上升的状态就可以构造出一个完整的山峰，其中我们默认山顶属于下降的序列，那么我们上升的长度和下降的长度是相等的，那么我们的数据范围就只有 50 了。

那么我们可以控制数组为 $f[i][j][k][u]$ 表示有 i 层，目前有 j 个上升的山坡，已经用了 k 个数字了，目前可以发展的不同优秀数列有 u 种。

那么我们枚举 $i + 1$ 的山坡个数 v，转移到 $f[i + 1][v][k + v][u * C(j - 1 + v, j - 1)]$。

因为如果要山峰不断掉，那么在 j 个山坡中的某两个山坡中有新的山坡那么前一个山坡就一定要上一层，那么就相当于有 j 个空格里放 v 个球，那么方案就是 $C(j - 1 + v, j - 1)$ 了。

那么我们就能直接计算答案了。

我们运用滚动数组来优化内存。

时间复杂度为 $O(\lfloor \frac{n}{2} \rfloor^5)$，空间复杂度为 $O(n^3)$。

来源

CF 301E

50 Context Advertising

题目大意

有一个文本，有 n 个单词，现在要把这个文本中一段连续的截取放到广告中去。

广告是 r 列，每列最多放 c 个字符，先把把一段文本放进去时不能把单词截断，而且每两个单词之间要空一个空格。

问你最多能放多少个连续的单词。

数据范围：$1 \leq n, r, c, r * c \leq 10^6, 1 \leq$ 总字符数 $\leq 5 * 10^6$

题解

我们先可以算出每个单词结尾的一行中最多第一个单词能是哪个单词。

然后就可以构成一棵树形结构，那么我们 dfs 搜下去的时候就可以直接递推计算出答案。

时间复杂度为 $O(n)$，空间复杂度为 $O(n)$。

来源

CF 309B

51 Escaping on Beaveractor

题目大意

一个 $b*b$ 的正方形在二维平面上。正方形里有 n 个箭头，平行于坐标轴。每对箭头是不会相交也不会互相接触。

当人到了某个箭头，人的方向会变成箭头的方向并移动到下一个箭头或者走出了校园。

每个时间单位走一个路程单位，可以认为没有障碍的。

有 q 个询问，每个询问为：

假设 0 时刻在 (xi, yi)，移动向量为 wi，计算时刻 ti 的时候人的位置。

数据范围：$0 \leq n \leq 100000, 1 \leq b, q \leq 100000, 0 \leq ti \leq 10^{15}$。

题解

此题很大地考验了代码实现能力。

此题可以分成 3 个小问题：

1：对于每个箭头算出方向上下一个箭头或者是出校园。

2：把每个箭头方向上下一个箭头设为这个箭头的父亲，那么构成了很多个环加外向树，把树上预处理倍增。

3：对于每个询问先找到路径上第一个箭头，然后倍增计算，如果倍增到了顶端的环上的话，就在环上二分。

我们一个一个问题来仔细分析一下。

对于第一个小问题：

这个问题核心思想是扫描线。

由于没有两个箭头会相交，那么我们只需要计算箭头的射出去的第一个点的那个方向上的第一个箭头是什么就可以了。而且做这个的时候我们可以把询问中找路径上第一个箭头一起做进去。

这个问题我们先处理所有方向是和 y 轴平行的询问。对于所有方向是和 x 轴平行的询问我们可以把坐标轴旋转 90 度做原问题。

现在可以把所有的箭头看作是线段。

然后我们以 x 轴开始做扫描线。我们现在已知 x。

我们维护一个 set，里面记录已知 x 的情况下，哪些 y 坐标有平行与 x 轴的线段。

我们还需要维护一个 set，里面记录已知 x 的情况下，所有和 y 轴平行的线段的两端的 y 坐标都放进去。

对于 x 固定的询问，有两种类型，和 y 轴的正反方向平行的询问。

设询问点为 (xi, yi)，我们就是需要知道 $yi \leq$ 或 $\leq yi$ 的 y 坐标。而这两个都可以在两个 set 中询问得到。

注意的是，在第二个 set 中，有可能这个线段包含了询问点，这个需要特判。

现在我们可以知道每个询问方向上的第一个线段的编号了。

对于第二个小问题：

我们对于每个未搜过的箭头开始搜索。我们先找到这颗树顶的环或者我们发现这棵树直接出了边界。

然后我们从树顶开始向下遍历预处理，计算出倍增的数组。

值得注意的是，我们某个箭头到它的父亲时，是到达它的父亲的中部的。

我们需要算出它到它父亲的点和它父亲的起点的距离。

然后我们把我们的路径转化成某个箭头到达它的父亲，然后走到它父亲的起点再继续出发。那么就变成了每个箭头的起点之间的距离关系了。

我们设中部到起点的距离为 pi，它到它父亲的距离为 hi。

很显然的是我们这个箭头的起点到它的父亲的起点之间的距离为 $hi - pi$。

但由于 $hi - pi$ 会是负数，我们就无法倍增了。

所以我们需要改变一下状态。

我们设某两个起点的距离为 $hfa - pi$。

如此每条边的边权就变成正数了，而他的意义则变成了：走了这条边，我们就可以到达它父亲的父亲。

那么我们只需要在初始出发的时候先减去 hi (先走一个父亲)，再在最后在特判一下，这个模型就可以工作了。

对于第三个小问题：

这个问题就只是各种特判了，只需要仔细分析就可以了，重要的内容已经在前两个问题中说明。

由此，此题便可以在**时间复杂度为 $O(nlog)$，空间复杂度为 $O(n)$**。

来源

CF 331D

52 Google Code Jam

题目大意

有一场 gcj，有 n 道题目，每道题目有一些性质：

1：每道题有 small 和 large。做出 small 得到 scoreSmalli 分，做出 large 得到 scoreLargei 分。

2：timeSmalli 表示做出 small 的时间，timeLargei 表示做出 large 的时间，每题必须先做出 small 再做 large。

3：做出 small 能保证一定正确，做出 large 有 probFaili 概率错误。

现在一共有 t 单位的时间，求用最优的策略做出期望最高的分数，如果期望分数一样再要求期望罚时最少。

注意罚时是最后一个提交正确解的时间。

数据范围：$1 \le n \le 1000, 1 \le t \le 1560, 1 \le scoreSmall_i, scoreLarge_i \le 10^9$

题解

显然可以发现的是对于某一种方案，一定是把 small 放前面，large 放后面，因为这样的分数期望是一定的，但罚时期望会小。

如果我们把题目按一定顺序排序之后，我们就可以直接 dp了。

对于一个题目我们可以枚举做 small 还是做 small 和 large。

用 $f[i][j]$ 来表示状态，到第 i 个题目，花了 j 个时间。

转移很显然。我们现在要求题目的顺序。

我们来计算罚时的期望。

我们设总时间为 T，现在有一个长 k 的题目序列，第一位为最后做，设失败的概率是 pi，设 $vi = ti * pi$。

罚时期望为 $T - p1 * t1 - p1 * p2 * t2 - ...$

那么我们用相邻两位翻转来计算比较器，那么我们可以用 $-vx - px * vy < -vy - py * vx$ 来排序。

然后直接做。

时间复杂度为 $O(n*t)$，空间复杂度为 $O(n*t)$。

来源

CF 277D

53 Candies Game

题目大意

有 n 个箱子，每个箱子有 ai 个糖果。

要求使得糖果都放到恰好两个箱子中，有一系列操作：每次选择两个不同的箱子 $i, j(ai \leq aj)$，然后从 j 中取 ai 个糖果放入第 i 个箱子里。

要求你构造一个操作序列使得满足要求，无解输出 -1。

操作序列长度不超过 10^6。

数据范围：$3 \leq n \leq 1000, 1 \leq \sum ai \leq 10^6$

题解

一开始如果只有一个数这是无解。

否则对于任意三元组 $i, j, k(0 < ai \leq aj \leq ak)$，如果我们能使得其中一个变空，那么此题就解决了。

我们定义一个运算使得 i, j, k 经过一些操作后 ai 变小，如果这个运算能完成，那么我们经过一定次数的递归后必定能使得 ai 变 0。

我们设 $x = \lfloor \frac{aj}{ai} \rfloor$，我们只需要把 $aj \to aj - x * ai$，我们的操作就完成了。

那么我们现在这样来操作：

我们把 x 二进制分开，因为每经过一个操作后，ai 都会变大一倍，那么我们就让 ai 不停地变大，然后如果 x 的某一位为 1，那么就让 (i, j) 操作一次，否则做 (i, k)。

由于我们是从低位到高位枚举的，所以能保证大小一定合法。

然后这样做一次运算后，aj 一定减小了至少一半，那么我们的操作数也有保证了。

时间复杂度为 $O(n * log * log)$，空间复杂度为 $O(n)$。

来源

CF 341E

54 Tournament-graph

题目大意

构造一个 n 个点的竞赛图，使得任意两个点的距离不超过 2。
竞赛图指的是任意两个点之间有一条有向边连接。
无解输出 -1。
数据范围：$3 \leq n \leq 1000$

题解

我认为只有一种构图方法。
先把 n 个点由有向边连成一个圈。
然后每次对于每个点，对于目前距离为 3 的连一条有向边。
如此我们可以发现当 n 是偶数时无解，否则就构出了。
时间复杂度为 $O(n^2)$，空间复杂度为 $O(n^2)$。

来源

CF 323B

55 Piglet's Birthday

题目大意

有 n 个架子，每个架子有 ai 个蜜罐，一开始全满。

有 m 个操作，第 i 个操作，会到第 ui 个架子，拿走 ki 个蜜罐，然后把蜜吃掉，把这些蜜罐放到第 vi 个架子上。

每次在一个架子上拿蜜罐的时候，会在所有蜜罐中随机挑选一个集合的蜜罐拿走。

问每个操作后，架子上的所有蜜罐都被吃完的架子期望个数。

数据范围：$1 \leq n, m \leq 10^5, 1 \leq ai \leq 100, 1 \leq ki \leq 5$

题解

可以发现的是每次拿走的蜜罐一定是空的，每个时刻每个架子的蜜罐个数一定是一定的。

那么我们可以用 $f[i][j]$ 表示第 i 个架子，还有 j 个蜜罐有蜂蜜的概率。

然后每次操作，只会修改一个架子的状态，转移显然。

那么**时间复杂度为** $O(n * 100 * 5)$，**空间复杂度为** $O(n * 100)$。

来源

CF 248E

56 Monsters and Diamonds

题目大意

有 n 种怪物，每次选一个怪物，会分成一定数量的怪物（可能为零）和至少一个钻石。怪物可能存在多种分裂方式。

一开始只有一只怪物，每次选中一个怪物，分裂，直到没有怪物为止，然后统计钻石的数量。

现在每种怪物至少有一种分裂方式，如果有多种方式，可以任意选一种。

问：对于每种怪物，确定以其为起始，可以得到的钻石最少最多分别是多少。

如果没办法分到没有怪物为止，输出 -1 -1。

否则如果能分完但可以得到无穷多的钻石，用 -2 代表无穷大。

如果有数字大于 314000000 （但不是无穷大），则用 314000000 代替。

数据范围：$1 \leq m, n,$ 分裂出的总数 $\leq 10^5$

题解

做这题需要想的很清楚。

我们先验证 -1，再验证 -2，再算 min。

对于验证 -1 时，我们先把可以不分裂出怪物的怪物标记，然后对于每种分裂方式，如果所有分裂出的怪物都被标记的话，那么被分裂的怪物也可以标记。

然后我们需要验证 -2，我们可以用记忆化搜索。

对于不是 -1 的怪物如果能形成环的话，就是 -2。

否则可以直接算出 max。

我们还需要计算 min。

计算 min，我们只需要验证 -1 的时候，每个时刻有很多分裂方式符合要求，我们把这些方式放进堆中，每次选择一个宝石数最小的先更新就可以了。

时间复杂度为 $O(n*log)$，空间复杂度为 $O(n)$。

来源

CF 325C

57 BerDonalds

题目大意

有 n 个点 m 条有权边的无向图，要求选择某个位置（可以是节点或是某条边上任意一点），使得到最远的点的距离最近。

数据范围：$1 \leq n \leq 200$

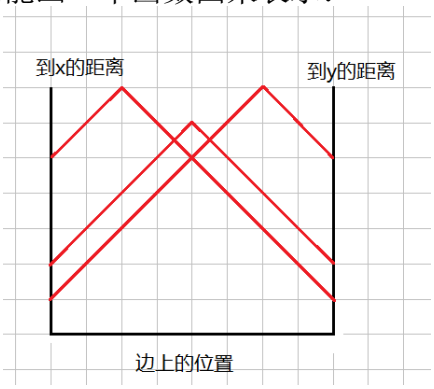
题解

经典算法。

先用 floyd 算出两两之间的最短路。

然后我们枚举边 (x, y)，对于其他的所有点都有两个值：到 x 的距离 lx 和到 y 的距离 ly。

现在我们设这个点在这条边上，那么对于位置在这条边上的所有位置，我们能画一个函数图来表示：



上图每条红线表示一个其他的点，现在就是对于每个可能的位置上方最上的位置进行计算答案。

我们可以发现这些红线的斜率是 1 和 -1。

由于我们求的是到最远的点的距离最近，那么我们就只要把每个可能成为答案的位置都比较一遍就可以了。

那么我们只需要把 lx 降序，如果现在的 ly 比以前的 ly 大的话，那么又构成了一个顶部的低谷，我们把相交的那个点再比较一下就好了。

时间复杂度为 $O(n^3 * log)$，空间复杂度为 $O(n^3)$。

来源

CF 266D

58 Fetch the Treasure

题目大意

有 h 个数，其中有 n 个数有值，一开始第 ai 个数值为 ci。

一开始能够到达的点为 $1, k+1, 2k+1, ...$

有 m 个 3 种操作：

1：增加一个 x，设此刻有 $a1, a2, .., ar$，那么能到达的数为 $1+\sum_{i=1}^{r} vi*ai$，vi 为非负整数。

2：把第 x 的数的值减少 y。

3：选择一个能到达的数，如果最大的值大于 0，那么把那个值输出，并把这个值清 0，否则输出 0。

数据范围：$1 \le h \le 10^{18}, 1 \le n, m \le 10^5, 1 \le k \le 10^4, 0 \le$ 第一种操作的数量 ≤ 20

题解

由于第一种操作的数量不超过 20 个，而且 k 不超过 1w。

那么我们可以通过 spfa 计算出到达所有模 k 为 x 的数至少为多少。

那么我们可以计算出能到达哪些数。

那么每次用堆来维护，直接做就可以了。

时间复杂度为 $O(n*log+k*20*20)$，空间复杂度为 $O(n)$。

来源

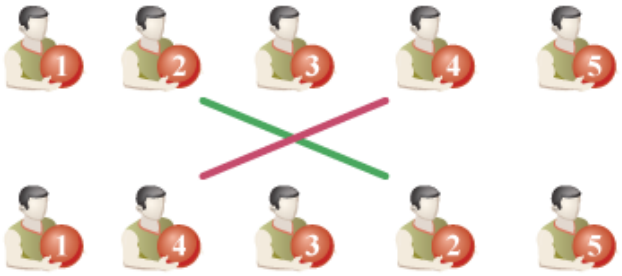
CF 311C

59 PE Lesson

题目大意

有 n 个学生，一开始第 i 个学生拿 i 号球。

现在有传球游戏，每次选择两个学生 (i, j)，把他们手中的球互换，如下图：



第 i 个学生最多传 bi 次球。

问经过传球游戏后有多少种不同的持球方案。

方案数模 1000000007。

数据范围：$1 \leq n \leq 10^6, 0 \leq bi \leq 2$

题解

对于一个排列，可以认为是很多个环。

对于一个环，如果有 n 个点，只有 2 个点交换了一次，$n-2$ 个点交换了两次。

我们可以计算出有 a 个能交换两次，有 b 个能交换一次。

我们先做能交换一次的人。

我们用 f_i 表示用了 i 个人的方案数。

转移是 $f_i = f_{i-2} * (i-1) + f_{i-1}$

然后对于能交换两次的，我们只需要每次枚举这个点把前面那个点交换就可以了，那么转移是 $f_i = f_{i-1} * i$

如此此题便解决了。

时间复杂度为 $O(n)$，空间复杂度为 $O(n)$。

来源

CF 316D

60 Two permutations

题目大意

有两个 n 的排列 p 和 q。

有 m 个询问，问在 p 中位置在 $[l1, r1]$，在 q 中位置在 $[l2, r2]$ 的数的个数。

强制在线。

数据范围：$1 \leq n \leq 10^6, 1 \leq m \leq 2 * 10^5$

题解

函数式线段树经典题目。

时间复杂度为 $O(n * log)$，空间复杂度为 $O(n * log)$。

来源

CF 323C

61 Pumping Stations

题目大意

给 n 个点 m 条边的网络流的图，求一个 n 个数的排列，使得相邻两点做源和汇做网络流，使得 $n-1$ 的流的和最大。

数据范围：$1 \leq n \leq 200, 1 \leq m \leq 1000$

题解

做此题需要介绍一下 $Gomory-Hu\ tree$。

这个数据结构就是我们建一个 n 个点的树，每条边有边权，任意两个点的最大流是树上路径的边权的最小值。

我们来构造这棵树。

我们把最大流看成最小割。

先任意选两个点做最小割。

最小割把 n 个点割成两个集合。

说明这两个集合点中的最小割是不会超过目前的割的。

然后我们两个集合递归做。

然后我们就计算出了任意两个点之间的最小割。

那么我们就可以来构树了。

每次我们选同一集合中的两个点，使得流最大，把它当树边，那么我们就只需要做最大生成树一样做就可以了。

那么对于题目中的答案，我们可以发现我要使得值尽量大的话，就是每次选一条边权最小的边，使得这条边只贡献一次答案，就是通过这条边把集合分成两个部分，分别递归做就可以了。

时间复杂度为 $O(n*maxflow)$，空间复杂度为 $O(m)$。

来源

CF 343E

62 Berland Traffic

题目大意

有 n 个点 m 条边的网络，1 为源，n 为汇。

每条边 (x, y)，设其流量为 t，容量为 c，如果 t 为正则表示从 x 流向 y，否则表示从 y 流向 x，且保证 $|t| \leq c$。

求最大流但要保证对于任意两个点 (x, y)，x 到 y 的流量并不会随着不同的路径而改变。（有可能有小于 0 的流量，流量的符号取决于 x 到 y 路径上这条边的方向）

求答案和方案。

数据范围：$1 \leq n \leq 100, 1 \leq m \leq 5000$

题解

需要保证那个特殊要求，我们只需要给每个点一个标号，然后每条边之间的边权都满足标号就可以保证了。

那么，我们发现标号相对的比例是一样的，那么我只需要构造出一组合法的标号，然后缩放就可以了。

那么我们把 1 号点标号为 0，n 号点标号为 1。用流量平衡来建方程，那么有 $n-2$ 个未知数和 $n-2$ 个方程，直接解出来就可以了。

时间复杂度为 $O(n^3)$，空间复杂度为 $O(n^3)$。

来源

CF 267C

63 Road Repairs

题目大意

有 n 个点 m 条边，边权为 0 或 1。
从 1 出发能到达整张图的最小权值和。
输出答案和方案。
数据范围：$1 \leq n, m \leq 10^5$

题解

最小树形图模版题。
此题数据比较水。
时间复杂度为 $O(n * m)$，空间复杂度为 $O(n * m)$。

来源

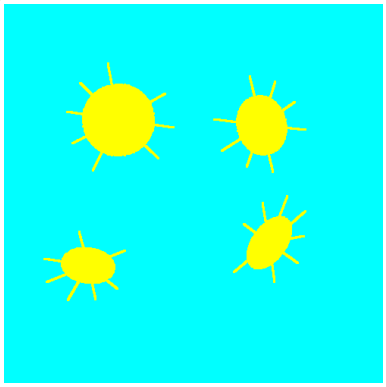
CF 240E

64 Suns and Rays

题目大意

一个 $n*m$ 像素的图形。

图形里有很多个太阳，太阳本体是旋转过的椭圆形，还有很多条射线，如图：



现在需要识别有多少个太阳和每个太阳有几条射线。

保证：

没有两个太阳有共同点，射线的宽度为 3 像素，太阳的轴的长度将介于 40 和 200 像素，没有两条射线相交，所有射线的长度将在 10 和 30 像素。

数据范围：$1 \leq n, m \leq 1600$

题解

太阳个数其实好判断，只需要计算连通块个数。

我们定义两个限制 lim 和 len。

计算射线的话，我们从每个点开始 bfs，当一层超过 lim 个点的话默认为已经不在射线上，那么退出，如果此层的编号超过 len 的话默认为射线，然后把遍历的点涂黑。

我们通过实际评测调试，使得 $lim = 5, len = 10$ 时能过此题。

时间复杂度为 $O(n*m*lim*len)$，空间复杂度为 $O(n*m)$。

来源

CF 316F

65 Levko and Game

题目大意

有 n 个点 m 条边的图，每条边有边权。

有两个人，一个人一开始在 s1，另一个人在 s2。

他们每个时刻走一个单位。

谁想到点 f，谁就赢，如果同时到达是平局。

现在有 k 条边，其中第一个人可以把每条边的边权改变成区间 $[li, ri]$ 中任意一个值。

问这个人用最优的方式改变边权，最后的结果是什么。

输出方案。

数据范围：$1 \leq n, m \leq 10^4, 1 \leq k \leq 100$

题解

首先我们可以发现改变的边权一定是区间的极值。

我们用 dij 来操作。

对于每个点，我们计算这个点是最优先被第一个人到达还是先被第二个人到达还是同时到达。

如果是先被第一个人到达，那么用最短的方法转移，否则用最长的方法转移。

这样做出来如果是 LOSE 其实并不是真正的 LOSE，有可能是 DRAW。

所以我们再做一遍，把边权改成：如果是先被第二个人到达，那么用最长的方法转移，否则用最短的方法转移。

这样此题就没有错误了。

时间复杂度为 $O(n * log)$，空间复杂度为 $O(n)$。

来源

CF 360E

66 Optimize!

题目大意

给一个长 len 的标准串 a。

再给长 n 的询问串 b。

问有多少个 b 中长 len 的子串能和串 a 相匹配。

如果说两个串能相匹配。

那么把两个串先升序排序 c 和 d，然后满足：$h \leq c_i + d_{len-i+1}$

数据范围：$1 \leq n, m \leq 150000$

题解

现在我们把 b 串升序排序。

那么我们就先把串 a，改变成每个数可以匹配 b 中的后 x 位。

那么对于 a 中长 len 的子串，我们把 xi 放入数列，求前缀和 gi，询问是否 $i \leq gi$。

那么就变成了线段树基础题。

时间复杂度为 $O(n * log)$，空间复杂度为 $O(n)$。

来源

CF 338E

67 Positions in Permutations

题目大意

一个 n 个数的排列的完美程度是指完美位置的个数。

如果第 i 位被成为完美，则需要满足 $|p_i - i| = 1$。

求完美程度为 k 排列个数。

答案模 1000000007。

数据范围：$1 \leq k \leq n \leq 1000$

题解

我们用 $f[i][j][4]$ 表示前 i 个数，完美的位置至少为 j 个，i 和 $i+1$ 这两个数是否选了。

转移很显然。

然后我们可以计算出完美的位置至少 j 的排列数。

那么我们可以通过容斥直接计算出有 k 个完美的位置的排列个数。

时间复杂度为 $O(n^2 * 4)$，空间复杂度为 $O(n^2 * 4)$。

来源

CF 285E

68 Tape Programming

题目大意

有一种编程语言，这个语言由数字和 < 和 > 构成的非空串。

程序运行有一个指针，最开始指针指向最左字符，向右移动。

重复以下操作直到指针指向串外：

1：如果指针指向的是一个数字，输出这个数字，指针向原来方向移动，那个数字减一，如果原来的数字为 0，则删除。

2：指针指的位置是 < 或 >，那么改变指针移动的方向，然后继续移动，但如果新的位置也是 < 或 >，则删除原来的字符。

现在有一个长 n 的串，有 m 个询问，每个询问把串的 $[li, ri]$ 看作是一个单独的程序，问每个询问每个数字会被输出几次。

数据范围：$1 \leq n, m \leq 10^5$，数字为 0 到 9。

题解

我们可以发现的是如果从左进入，再出去就结束了。

那么我们只需要从第一个位子开始模拟，如果从左出去了，那么把目前遍历的字符删掉，继续模拟，如果从右出去了，那么就结束了。

我们把每一步的操作都记录下来，用前缀和计算。

现在我们有询问，我们可以找到前面我们模拟中第一次到达的这个点的时候，和第一次出去这一段的边界。那么我们只需要直接算答案就可以了。

时间复杂度为 $O(n * 10 * 10)$，空间复杂度为 $O(n * 10 * 10)$。

来源

CF 238D

69 The Great Julya Calendar

题目大意

一开始给出一个数 n。

每次可以把这个数减去这个数其中一个数字，直到变成 0。

求最少的操作数量。

数据范围：$0 \leq n \leq 10^{18}$

题解

显然是事实是每次减去最大的数就能保证次数最少。

我们定义二元组 (x, y) 表示尾数为 x，前面的数字最大为 y的数，最少几次能把尾数减成负数。

那么我们可以记忆化搜索。

转移就是把尾数再减小一位。

由于如果减成负数的话，那么每一个长度的数都只有 100 个状态。

那么**时间复杂度为** $O(18000)$，**空间复杂度为** $O(18000)$。

来源

CF 331C

70 Polo the Penguin and Lucky Numbers

题目大意

定义幸运数字是由 4 和 7 组成的数字。
现在询问两个长度相同的正整数 l 和 r。
设在 l 和 r 中有 n 个不同的幸运数字，升序排列。
计算：$\sum_{i=1}^{n-1} a_i * a_{i+1}$
答案模 1000000007。
数据范围：$1 \leq l \leq r \leq 10^{100000}$

题解

数位 dp。
我们用 $f[i]$ 表示长 i 位的幸运数字乘积和。
我们转移的时候，可以先把首位加一个 4 和加一个 7 的乘积算出来。
然后把中间的两个算出来再乘一下，加进去转移到 $f[i+1]$。
转移的过程中，我们只需要把式子拆出来就可以了。
现在询问的过程中，经典问题。
时间复杂度为 $O(n)$，空间复杂度为 $O(n)$。

来源

CF 288E

71 Race

题目大意

有一个 $n*m$ 的矩阵表示城市。

一个交叉路口占据一个格子，每条道路是水平的或竖直的。

从每个交叉路口到相邻的格子会 1 个时间，道路上的每个点都有到相邻格子的时间。

现在有一个路径，经过交叉路口的顺序，现在问你现在在第 k 个时刻在哪个位子。

有 26 个交叉路口。

数据范围：$3 \leq m, n \leq 1000, 1 \leq k \leq 100000$

题解

直接暴力维护每对交叉路口的时间，直接做。

时间复杂度为 $O(26*26*n*m)$，空间复杂度为 $O(n*m)$。

来源

CF 241F

72 Cubes

题目大意

有一个 $n*n$ 的矩阵，每个矩阵是一个单位的正方形，边与坐标轴平行，一对角为 $(0,0)$ 和 (n,n)，每个正方形有一个高度。

现在在无穷远的地方，以向量 $v=(vx,vy,0)$ 看矩阵，问能看到几个不同的立方体。

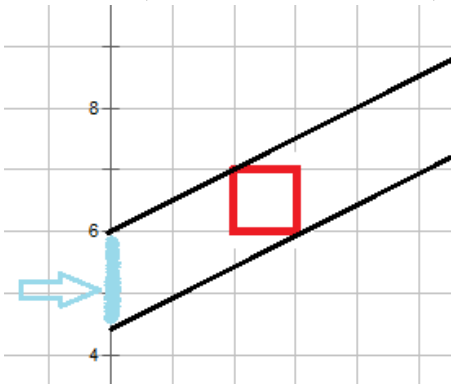
如果一个立方体是可见的，那么属于立方体上存在一个点 p，从 p 以向量 -v 发出射线上不存在属于其他立方体的点。

数据范围：$1 \leq n \leq 1000$

题解

先把 -v 转成第一象限的向量。

可以把一个立方体如图转成一个投影在 y 轴上的区间：



那么我们只需要把立方体按右上程度排序，用高度进行线段树操作。

然后就变成线段树区间覆盖。

时间复杂度为 $O(n^2*log)$，空间复杂度为 $O(n^2)$。

来源

CF 243D

73 Meeting Her

题目大意

有 n 个点 m 条有向边的图，所有边等长。

现在人在 a，他想坐公交车去 b。

有 k 个公交车，每个时刻第 i 辆车随机选择一条从 si 到 ti 的最短路，然后通过。如果没有路径，那么就不开。

如果一辆车通过人所在的点，那么可以上车再在中途下车。

可以上车然后直接下车。

求最坏的情况要做几次公交车。

数据范围：$2 \leq n, k \leq 100, 0 \leq m \leq n * (n-1)$

题解

我们此题采用无限更新方式。

对于每辆车，我们先把可能走的点和必经的点预处理出来。

我们预处理用 floyd 来做。

我们用 fi 表示 位置 i 最坏要做几次公交车能到达终点。

然后每次我们用每辆车更新答案。

对于每辆车我们枚举必经的点用预处理的图记忆化搜索更新。

如果 k 辆车做完都不能使得 f 更新，那么就结束了。

不然继续枚举每辆车更新。

时间复杂度为 $O(n^4)$，空间复杂度为 $O(n^2)$。

来源

CF 238E

74 Two Sets

题目大意

有 n 个数，要使得这 n 个数分成两个集合，把两个集合的数 xor，算出 x1 和 x2。

求 $max(x1+x2)$，如果一样大，则使得 x1 尽量小。

数据范围：$1 \leq n \leq 100000, 0 \leq$ 数 $\leq 10^{18}$

题解

把所有数 xor 得 V。

题目使得求一个数 x2，使得 $max(x2+(x2\hat{}V))$ 并尽量让 x2 大。

我们可以把位重排列。

可以发现如果 V 的某位为 0，则这类的位比为 1 的要重要，因为如果 x2 这位是 1 的话，x1 这位也是 1。

如果一样的类按原来的高低定义重要性。

然后我们用位数高斯消元来求出最大的 x2。

那么我们在保证 x2 最大的同时也保证了答案最大。

时间复杂度为 $O(n*log)$，空间复杂度为 $O(n)$。

来源

CF 251D

75 Jeff and Removing Periods

题目大意

有一个长 n 的序列 a，执行下列操作：

1：选择 v, t, k，要求满足 $a_v = a_{v+t} = a_{v+2*t} \ldots = a_{v+k*t}$

2：把这 $k+1$ 个数删除。

3：重新排列这个 $n-k-1$ 个数。

代价就是把所有数字删除的次数。

现在有 m 个询问，每次询问把 a 中的 $[l, r]$ 的代价。

数据范围：$1 \leq n, m \leq 10^5$

题解

此题就是计算一个区间中不同数的个数和是否有一个数在一个区间中下标是等差数列。

很简单，只需要离线和树状数组维护就可以了。

时间复杂度为 $O(n * log)$，空间复杂度为 $O(n)$。

来源

CF 351D

76 Close Vertices

题目大意

有 n 个点的树，定义路径的长度为边的个数，路径的权重为边的权值和。

问有多少对点之间的路径的长度小于等于 L 且权重小于等于 W。

数据范围：$1 \leq n \leq 10^5$

题解

点分裸题。

时间复杂度为 $O(n * log^2)$，空间复杂度为 $O(n)$。

来源

CF 293E

77 Pilgrims

题目大意

有 n 个点的树，其中有 m 个点上有修道院，里面有一个人。

每次每个人都会选择一个最远的修道院走过去，如果有多个，随机选一个。

现在可以摧毁一个点，如果使得有一个人无法到达所有最远的修道院的话，那么就不开心。

求最多能让几个人不开心。还要计算有多少种方法。

数据范围：$1 \leq n \leq 10^5$

题解

每个人可以确定一条链，如果链上摧毁一个点，那么那个人就不开心了。

这个确定链，我们可以用树形 dp 和换根计算。

然后就是链加，找到点权值最大的点和个数。

时间复杂度为 $O(n)$，空间复杂度为 $O(n)$。

来源

CF 348E

78 Flights

题目大意

有 n 个点，有 m 条单向边，一开始每条边边权为 1。

现在要把一些边的边权改成 2，使得从 1 到 n 的所有路径的边权和相等。

如果有解输出方案。

边不会形成环，两个点直接不会多于一条边。

数据范围：$1 \leq n \leq 1000, 1 \leq m \leq 5000$

题解

此题就是给每个点一个标号 ai。

对于一条边 (x, y)，要求满足 $ay - ax \leq 2, ax - ay \leq -1$。

$a1 = 0$，求 an 的最短路。

做差分约束系统，如果 1 需要调整了就是无解。

由于此图是拓扑图所以没有负环。

时间复杂度为 $O(m)$，空间复杂度为 $O(m)$。

来源

CF 241E

79 Greedy Elevator

题目大意

有 m 层，有一个电梯。

一开始电梯在第一层，每个时刻可以上升或下降一层或不动。

有 n 个人，每个人在 ti 时刻到 si 层等电梯，要到 fi 层。

每个时刻对于每个人，我们计算要向上需求的人数 p_{up} 和向下需求的人数 p_{down}。

如果 $p_{down} \leq p_{up}$ 那么电梯上升一层，否则下降一层。

问每个在哪个时刻到达所需的层。

数据范围：$1 \leq n \leq 10^5, 2 \leq m \leq 10^5, 1 \leq ti \leq 10^9, si \neq fi$

题解

按照时间用树状数组模拟。

时间复杂度为 $O(n * log)$，空间复杂度为 $O(n)$。

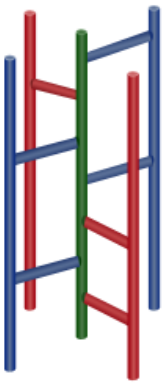
来源

CF 257E

80 Wall Bars

题目大意

现在需要设计一种玩具，如图：



我们定义中间那个管子高度为 n。

在高度为 $1, 2, 3, .., n$ 的地方，恰好有一根水平的杆子在 4 个方向的其中一个。

如果两根杆子的距离不超过 h，且方向相同，那么就可以从一根杆子爬到另一根上。

一开始可以爬到高度在 $[1, h]$ 之间的杆子上。

从一开始能到达高度在 $[n - h + 1, n]$ 的杆子算成功。

求不同的方案数。

答案模 1000000009。

数据范围：$1 \leq n \leq 1000, 1 \leq h \leq min(n, 30)$

题解

我们用 dp 来解决此题。

我们一开始用 $f[i][j][k][v][u][r]$，表示高度为 i，目前这根杆子是第 u 根，r 表示是否能由下层爬上来，另外 3 根杆子和高度为 i 的差距分别为

j, k, v。

状态数为 $1000 * 30 * 30 * 30 * 4 * 2$，转移为枚举下一根是哪一根。

然后我们来优化这个算法。

首先，我们发现是不需要记录 u，因为所有方案中第 i 层是哪一根都是可能的，那么我们把答案除 4 后再枚举这根和下一根就可以了。

但这个只缩小了内存，并没有改进复杂度。

但是我们可以发现转移的时候，其实并不需要把答案除 4，只需要默认这个是第一根就可以了，因为这样代表的方案数性质是一样的。

最后我们采取滚动数组。

那么**时间复杂度为** $O(n * 30^3 * 2 * 4)$**，空间复杂度为** $O(30^3)$。

来源

CF 268D

81 Rats

题目大意

现在有一个 $n*m$ 的图。

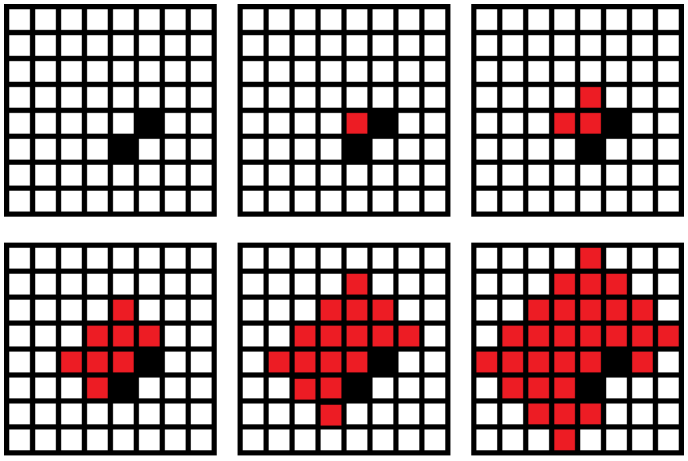
格子分为墙壁、老鼠和空地。

现在你可以在两个非墙壁的位置引爆两个手榴弹来灭鼠。

目的是把所有老鼠灭掉。

一个手榴弹的引爆是有范围的。

每个时刻向外扩展一圈，扩展 k 个时间点，如下图：



求一对合法的放手榴弹的位置。

数据范围：$1 \leq n, m \leq 1000, 1 \leq k \leq 8$

题解

由于 k 比较小，我们可以发现一个空地，爆炸一个手榴弹只会扩展不超过 150个点。

那么如果老鼠个数超过 300 个，就是无解。

我们枚举每个老鼠，计算出每个老鼠能被多少个格子炸掉。

我们枚举把第一个老鼠炸掉的位置，再枚举一个位置，再判断。

时间复杂度为 $O(150^3)$，空间复杂度为 $O(n^2)$。

来源

CF 254D

82 Maximum Waterfall

题目大意

现在要建一个人造瀑布。

顶部高度为 t。

中间有 n 块板嵌在上面。每块板可以看成一条水平的线段，从 (hi, li) 到 (hi, ri)。

顶部可以看作是 $(t, -1e9)$ 到 $(t, 1e9)$。底部可以看作是 $(0, -1e9)$ 到 $(0, 1e9)$。

如果水可以从第 i 块板流到第 j 块板，需要满足：

$max(li, lj) < min(ri, rj), hj < hi$

不存在 k 且 (i, k) 和 (k, j) 均满足以上条件。

而且水的流量是 $min(ri, rj) - max(li, lj)$。

一条从顶到底的瀑布的水流量为，相邻两个板的水流量的 min。

求最大的瀑布的水流量。

没有两块板有重叠的部分。

数据范围：$1 \leq n \leq 10^5, 2 \leq t \leq 10^9$

题解

我们从低到高枚举板，我们用 fi 表示第 i 块板开始的瀑布最大的水流量。

每次加一个板，我们把这个板和下面能直接流到的板更新答案。

我们维护用 set 一下每一层垂直能到的所有板。

对于每个区间我们记录左端点。

每次加一块板，我们把中间的 set 都去掉，后面再加上一个。

注意更新的时候，我们需要判断是否这两块板中间没有别的板了。

时间复杂度为 $O(n * log)$，空间复杂度为 $O(n)$。

来源

CF 269D

83 Doodle Jump

题目大意

现在有 n 个平台，第 i 个平台的高度是 $i * a\% \, p$，a 和 p 是互质的正整数。

现在把平台按照升序排序，如果相邻两个平台的高度差不超过 h 的话，表示可以跳上去。

问是否能从高度为 0 的地面跳到最高的平台上。

有 t 组数据。

数据范围：$1 \leq t \leq 10000, 1 \leq a \leq 10^9, 1 \leq n < p \leq 10^9$

题解

我们主要是求最大的间隔。

对于一组数据。

$n = 16, a = 7, p = 25$

我们把跳的高度写出来，每次模了换一行，如图：

```
0 7  14 21
3 10 17 24
6 13 20
2 9  16 23
5 12
```

我们可以发现每一列的间距都是恒定的，当然如果是最后一行的末尾不够长的话，就用最短的那列，因为这样造成的最大的间隔最大。

而且那些加上 a 要模的那些都是没用的，因为最后一列的数只是前一列数的最小的一部分。

那么就变成原问题了，只是 $a \to (a - p\% \, a), p \to a, n \to \frac{a*n}{p}$，其中 n 还是要分情况讨论的。

但是这样递归是不行的。

因为当 $a = n = p - 1$ 的时候复杂度是 $O(p)$的。

其实把 a 变成 $min((a-p\%\ a),(p\%\ a))$ 就完成了。

因为如果 $a*2>p$，我们可以认为板是从上往下放的，所以 a 和 $p-a$ 是等价的。

时间复杂度为 $O(t*log)$，空间复杂度为 $O(1)$。

来源

CF 346E

84 Colorful Stones

题目大意

有两个字符串 s 和 t，每个串都由 R, G, B 三种颜色构成。

一开始两个人分别在两个串的头。

现在有很多个命令，每次一个命令为三种颜色的其中一种，如果此刻有人的位置的字符和命令一致，那么向后走一步。

要求不能走出字符串。

以两个人的位置构成了状态，问有多少中不同的状态。

数据范围：$1 \leq |s|, |t| \leq 10^6$

题解

我们对于第一个人的位置确定求第二个人可能的位置。

我们先计算出固定第一个人的位置，第二个的人能走出的最近和最远的距离。

但这中间还是会有不能走到的状态。

如果我现在第一个人前面有连续 x 个相同的颜色，那么第二个人前面如果和第一个人颜色相同且前面连续相同颜色的个数小于 x 的话，那个位置是不能达到的。

那么我们只需要用预处理就可以解决了。

时间复杂度为 $O(|s|)$，空间复杂度为 $O(|s|)$。

来源

CF 264D

85 k-Maximum Subsequence Sum

题目大意

有一个长 n 的数列，有两种操作：

1：把 ai 改为 p。

2：在 $[l,r]$ 中取不超过 k 段的最大不相交的子段和。

数据范围：$1 \leq n,m \leq 10^5, 1 \leq k \leq 20$

题解

我们可以把不超过 k 段的最大不相交的子段和看成网络流。

每次流把一段选中，然后把这段翻转。

那么我们只需要做 k 次，求一个权值和最大的子段，选中，然后翻转。

时间复杂度为 $O(n*log*k)$，空间复杂度为 $O(n*log)$。

来源

CF 280D

86 Matrix

题目大意

有一个 $n*n$ 的 01 矩阵，现在要把列重新排列使得每行的 1 连续。

求某个方案或表示无解。

数据范围：$1 \leq n \leq 500$

题解

题目可以简化成有 n 个限制，每个限制要求把一个集合连续在一起，求一个排列方案能满足所有限制。

我自己发明了一个做法，但后来和别人交流的时候发现这个算法叫 pq tree，已经存在。

下面我来介绍一下我的做法：

做法的核心思想是搞一个树形的结构，现在我用 () 表示一个节点，它的儿子可以乱选，[] 表示它的儿子必须按顺序，用 . 表示叶子。

对于 []，我们还需要记录这个点是否可以翻转 flag。

一开始我有一个 () 的根，把 $1, 2, ..., n$ 所有叶子都作为这个点的儿子接在下面表示开始时所有都可以乱选。

然后对于每个限制我都修改这棵树，使得它满足限制，最后求答案的时候我只需要遍历一遍这棵树就可以了。

接下来我介绍一下修改这棵树的方法。

一开始我运用 dp 遍历整棵树来计算哪些点的子树叶子对于我目前的限制是全集或空集或子集。

然后我从根开始搜下去。

设当前点为 i。

我们搜的时候也有 3 种限制：限制靠左或靠右或无要求。

如果是全集或空集那么退出。

我们先分析 () 的情况：

我们遍历 i 的儿子，对于 i 的儿子我们以全集或空集或子集分成三个集合 a, b, c。

如果 c 的大小超过 2，无解。

如果 c 的大小等于 2，但要求靠左或靠右，无解。

然后我们继续搜索 c 集合。

有以下几类情况：

1：a 为空，c 只有一个元素且无要求，则是无要求。

2：c 只有一个元素且有要求，那么按照要求。

3：c 只有一个元素且无要求，那么我们默认向右并。

4：c 有两个元素，一个向右一个向左。

接下来我们考虑 a 集合。

首先我们新建一个 () 为 x，把 a 集合的元素连进去，因为全集所以可以随便选。

我们再新建一个 () 为 y，把 b 集合的元素连进去，因为空集所以可以随便选。

然后我们根据 c 的大小分类讨论，我们需要建立一个点 z 来包括 c。

1：如果 c 是空集的话，则 z 可以无视。

2：如果 c 只有一个元素，则把 z 和 x 连在一起再建一个新的点 [] 来固定，如果有要求就按照要求放位置，否则我们可以发现这个 [] 是可以翻转的。

3：如果 c 有两个元素，那么就是靠右的放左，靠左的放右，中间放 x 来建一个点 [] 固定，显然这个翻转也是可行的。

最后还有 y 要处理，如果无要求的，y 是不应该存在的，必须拆掉，否则按照要求放。

如此这一类的情况就解决了。

但实现的时候还需要注意的是如果只有一个儿子，那么需要把这个点去掉，不然不能保证每个时刻点数是 $O(n)$ 级别的。

接下来我们来分析 [] 的情况：

可以发现 i 的儿子不是空集一定是连续一段的，那么我们先找到那一段连续的，如果有多段是无解。

对于那一段只能是两边是子集，中间是全集。

如果是无要求，那么我们继续搜索使得左部是靠右的，右部是靠左的，如果只有一个元素，那么也是无要求的。

如果有要求的话我们需要判断，左边或右边是否靠在边界和是全集，如果只有一个元素那再简单地特判一下就可以了。

不过目前无解并不是真的无解。

因为在分析 () 的时候我们已经发现了有些 [] 是可以翻转的，那么我们发现无解的时候就只需要翻转再验证一遍。

如果有要求的话，那么 i 这个点就不能再翻转了。

这样 [] 就简单地分析完了。

现在我们再来分析一下翻转的程序。

对于翻转的话，我们只需要打一个标记，搜的时候下传，然后只需要搜索 []，而且是那些不能翻转的。

如此此题就搞定了。

我们来计算一下复杂度，可以发现一共有 n 个叶子，由于我们控制的是每个非叶子点儿子个数必须大于 1，那么，最坏的情况可以按照线段树，所以内存是 $4*n$ 个，但由于一些无用的开内存，所以我们把内存开到 $n*8$ 是比较保险的。每次限制我都遍历一遍整张图，那时间复杂度也可以保证了。

时间复杂度为 $O(n^2)$，空间复杂度为 $O(n)$。

来源

CF 243E

87 The Evil Temple and the Moving Rocks

题目大意

有一个 $n*n$ 的矩阵。

有一些空地上有石头，石头有 4 种类型：$\hat{},<,>,v$，表示为向上、左、右、下移动。

现在选中一个石头，然后这个石头向本身的方向移动。如果撞到了另一个石头，那么这个石头停止移动，另一个石头开始向本身的方向移动。如果撞到了边界就结束了。如果撞的次数超过 1000w 次也结束。

如果一个石头改变了原来的位置然后撞到了别的石头或边界就发出一个声音，要使得声音次数到达或超过 x 次。

求构造这个方案。

数据范围：$n=100, x=10^5$

题解

构造题。

我的方法是每两行为一块，如图：

>>>>> . > . > . > . > .v

$\hat{}$. < . < . < . <<<<<<<<

然后一开始选中右上角。

然后把这些块复制到满。

可以发现，如果一块走完了，就会到达下一块。

此种构造方法大概能到达 16w 左右。

时间复杂度为 $O(n^2)$，空间复杂度为 $O(1)$。

来源

CF 329D

88 Ksusha and Square

题目大意

有一个面积不为 0 的 n 个点的凸多边形。

求如果选两个不同的格点（可以在凸多边形内部或边上），然后以他们之间的连线为对角线做一个正方形，正方形期望面积是多少？

凸多边形没有 3 点共线。

两个不同的格点顺序不一样算同一种。

数据范围：$3 \leq n \leq 10^5, |\text{坐标范围}| \leq 10^6$

题解

设对角线是 z，正方形的面积为 $\frac{z^2}{2}$，而且 $z^2 = x^2 + y^2$，x, y 为坐标的差。

那么我们就可以把 x 轴和 y 轴分开考虑了。

然后我们可以通过扫描线算出每个 x 和每个 y 各有几个格点。

然后拆公式直接算。

时间复杂度为 $O(10^6)$，空间复杂度为 $O(10^6)$。

来源

CF 293D

89 White, Black and White Again

题目大意

有 n 天，有 w 件两两不同的好事和 b 件两两不同的坏事。

每天会全部发生好事和全部发生坏事，至少发生一件事。

现在这个 n 天，会先发生一些天的好事再发生一些天的坏事再发生一些天的好事（一些 > 0）。

输出方案数。

答案模 1000000009。

数据范围：$3 \leq n \leq 4000, 2 \leq w \leq 4000, 1 \leq b \leq 4000, n \leq w + b$

题解

我们枚举坏事发生的天数，然后直接用组合数计算方案数。

时间复杂度为 $O(n^2)$，空间复杂度为 $O(n^2)$。

来源

CF 306C

90 Buy One, Get One Free

题目大意

有 n 个物品，每个物品有一个价格 ai。

买一个物品可以选择一个价格严格低于买的物品的物品进行赠送。

问最少花多少钱能全部买走。

数据范围：$1 \leq n \leq 10^5, 1 \leq ai \leq 10^9$

题解

$O(n^2)$ 的 dp 显然。

我们把相同价格的物品合并。

用 $f[i][j]$ 表示状态，从价格高到低，买了前 i 个物品，有 j 个是免费的最多省了多少钱。

现在我们来优化这个状态。

我们固定 i，把 j 差分。

我们只需要维护那个差分的值就可以了，我们可以发现差分的值一定是递减的，那么我们用堆维护。

我们可以发现如果固定 i，免费拿的物品最多的个数是固定的，那么我们堆中维护的差值的个数也是要固定的。

每次我们加一个价格的物品。

如果前面还有可以带免费的，那么我们就先把物品放到免费里面去。

否则我们考虑用我们这个物品来替换堆中原来的差值，一个差值可以代表一个物品。

我们设差值为 x。

如果 $x \leq ai$，那么我们可以发现，我们可以把 x 不选，而使得前面多出一个物品，那么就能用两个 ai 来替换了，因为前面物品也能带一个免费的。

如果 $ai < x$，那么如果我们把用两个 ai 替换前面的话，差值就变为 x 和 $2 * ai - x$ 了，表示如果选一个就是 x 或 $2 * ai - x$，如果选两个就是两个 ai。

很显然的是我们的 x 一定是从小到大替换的。

但是如果我们还剩下一个 ai 的话，如果我们能替换堆中剩余一个差值使得答案变优，那么替换，前面不能这么做，因为我们要保证差值最多，这个是必须要固定的，只剩下一个 ai 的时候，我们无法使得差值个数增多，那么才可以替换原来的值。

最后我们只需要把差值为正的都取出来就答案了。

时间复杂度为 $O(n * log)$，空间复杂度为 $O(n)$。

来源

CF 335F

91 Balance

题目大意

有 n 个容器，有 m 条管道连接两个容器，每个容器容量为 v，任何时刻都不能超过 v。

现在每个容器初始水量是 ai 目标水量是 bi。

求一个方案使得完成所有目标，一个方案的步骤数量不能超过 $2*n^2$。

数据范围：$1 \leq n \leq 300, 1 \leq v \leq 10^9, 0 \leq m \leq 50000$

题解

对于每个用管道连通的连通块的初始水量和目标水量和要相等，否则无解。

现在每个连通块分别做。

我们把每个连通块变成一棵树的形态，然后不停地删叶子，使得叶子满足目标，每次的步骤不会超过 n，那么就可以构造解了。

时间复杂度为 $O(n^2)$，空间复杂度为 $O(m)$。

来源

CF 317C

92 Mystic Carvings

题目大意

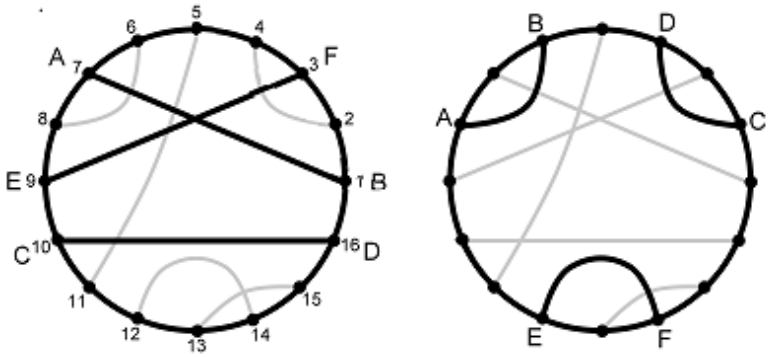
有 $2*n$ 个点平均分布在一个圆形的地域的边缘处，有 n 条路径，每条路径连接两个点，有个点只有一条路径。

现在要在 n 条路径中选 3 条，选 6 个点。

每条路径都有一个距离值：把这两个点在边缘走至少需要经过选中的点的数量。

如果每条选中的路径的距离值都一样则说这个方案是公平的。

如图，第一张图是不公平的，第二张图是公平的：

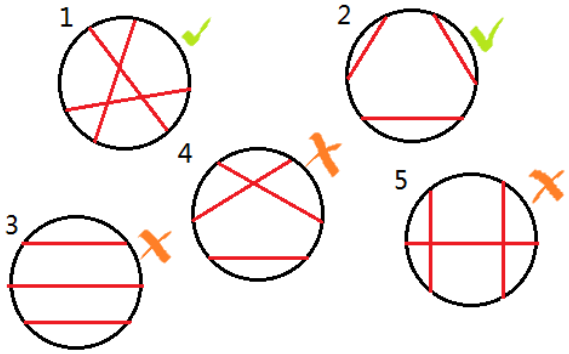


问有多少种本质不同的方案。

数据范围：$1 \leq n \leq 100000$

题解

路径更加相交的情况分类，一共有 5 种情况：



我们可以通过减去无解的方案来计算答案。

我们计算 3，我们枚举中间的那条路径就可以保证方案唯一了。

对于 4 和 5，我们可以枚举 5 中一条竖的路径，然后一条穿过，一条不相交，然后可以发现方案数算了两遍。

那么此题就可以用扫描线和树状数组来完成了。

时间复杂度为 $O(n*log)$，空间复杂度为 $O(n)$。

来源

CF 297E

93 Shaass and Painter Robot

题目大意

有一个 $n*m$ 的网格。

一开始在边界上选一个点，然后给一个方向（左上，左下，右上，右下之一），然后不停地走，如果撞到边界，那么遵循光的反射定律改变方向，然后继续走。

每走一格涂黑一格，如果涂过还是要涂，如果在某个时刻整个网格黑白相间，那么结束。

问是否能结束，如果能，输出走的格子数量，否则输出 -1。

数据范围：$1 \leq n, m \leq 10^5$

题解

转折点一定在边界上，我们模拟这个过程，如果走了 100w 步没有把该走的边界走完就是无解。

我们可以发现结束一定是在边界上的，所以我们就只需要记录边界上还有多少个点还没走到，当走完了输出答案。

时间复杂度为 $O(10^6)$，空间复杂度为 $O(1)$。

来源

CF 294D

94 Ciel and Flipboard

题目大意

有 $n*n$ 的数字矩阵。

n 是奇数，设 $x=\frac{n+1}{2}$，每次可以选择一个 $x*x$ 的子矩阵，然后把数字乘 -1。

可以操作任意次。

求最大化数字和。

数据范围：$1 \leq n \leq 33$

题解

我们定义 $f[i][j]$ 表示 (i,j) 这个格子乘 -1 的次数奇偶。

由于 $x=\frac{n+1}{2}$，所以可以得到一系列等式：

$f[i][j]\hat{}f[i][x]\hat{}f[i][j+x]=0, j \leq x$

$f[i][j]\hat{}f[x][j]\hat{}f[i+x][j]=0, i \leq x$

那么，我们可以发现，如果我们把 $f[1..x][x]$ 和 $f[x][1..x]$ 确定的话，那么其他的每 4 个数字就是有关联的，那么我们可以贪心算出答案。

但如果这样的话我们需要枚举 $2*x$ 位。

现在我们改进算法，我们只枚举 $f[1..x][x]$，那么我们发现 $f[x][1..x]$ 是互相不影响的，那么我们可以一位一位确定。

这样我们就可以解决此题了。

时间复杂度为 $O(2^x*n*n)$，空间复杂度为 $O(n*n)$。

来源

CF 321D

95 The Last Hole!

题目大意

平面上有 n 个圆，第 i 个圆圆心在 (xi, yi)。

一开始每个圆半径为 0，现在所有圆半径同时开始变大，时刻 t，圆半径为 t，随着圆增大，圆会相交。

某些时刻很多圆之间会构成一个洞，现在问最后一个洞消失的时间。

如果没有洞构出过，那么输出 -1。

数据范围：$1 \leq n \leq 100$

题解

我们可以发现洞本质是 3 个圆构成，但如果 3 个点构成的是直角三角形，那么是不会构成洞的。

但如果是矩阵，还是会构成洞的，那么我们只需要把所有洞最后消失的时间取一个 max，就可以得到答案了。

3 个圆的洞结束的点是外心，4 个圆是对角线的交。

时间复杂度为 $O(n^4)$，空间复杂度为 $O(n)$。

来源

CF 274C

96 Have You Ever Heard About the Word?

题目大意

一个重复串的定义是这个串是由一个串连接自身形成。

给一个长 n 的字符串，每次选一个最短的子串是重复串，如果多于一个，选最左边那个，然后把一半删掉，问最后的串。

数据范围：$1 \leq n \leq 50000$

题解

我们可以找到所有的子串重复串，然后把后一半标记，最后输出所有没标记的字符。

我们可以枚举重复串一半的长度，然后把串根据长度分段，相邻两个分割点，我们二分前面后面最长公共长度，然后可以计算所有可能的重复串。

时间复杂度为 $O(n * log * log)$，空间复杂度为 $O(n)$。

来源

CF 319D

97 Sereja and Squares

题目大意

有 n 个坐标，第 i 个是 $(i, 0)$。

现在给每个坐标一个小写或大写英文字母，但不用 x, X。

如果一个方案是漂亮的，要满足一下性质：

所有点分为若干对，每个点属于一个对。每对点中横坐标小的需要是小写，大的是大写，这两个点之间的线段当直径做圆，要使得没有两个圆会相交。

现在有一些坐标的字母不知道，问所有漂亮的方案数。

答案模 4294967296。

数据范围：$1 \leq n \leq 10^5$

题解

用 $f[i][j]$，表示到第 i 个点，前面有 j 个点需要配对。

转移显然。

我们运用滚动数组和卡常 数，就可以过掉此题。

时间复杂度为 $O(n^2)$，空间复杂度为 $O(n)$。

来源

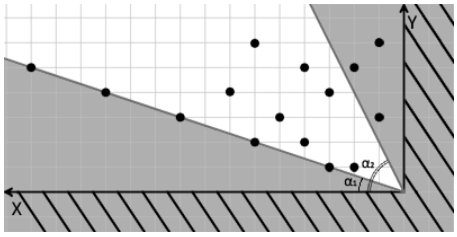
CF 314E

98 Donkey and Stars

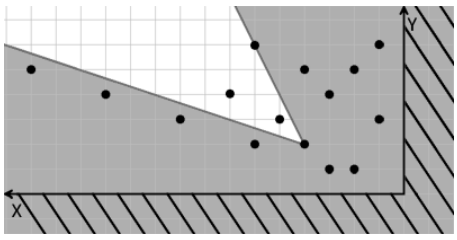
题目大意

平面上有 n 个不同位置的星星 (x, y)，人在原点。

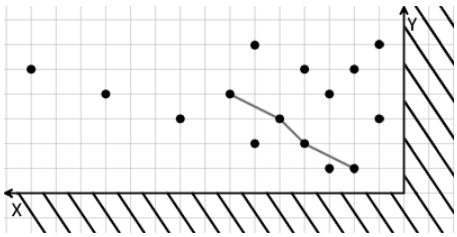
人的视野是一个两条射线之间的区域，如图：



每次都可以在视野中选一个星星（不包括边界），然后再从那个星星出发，如图：



最后得到了一条星星的链，如图：



问最长的长度是多少。

数据范围：$1 \leq n \leq 10^5, 0 \leq x, y \leq 10^5$

题解

我们可以把视野的两条射线分开考虑。
然后把每个星星经过这两条射线投影到 y 轴上，得到两个值。
然后把每个星星的坐标用那两个值重构。
这个星星出去的视野可以表示为一个矩阵了。
那么直接扫描线加树状数组就可以维护了。
时间复杂度为 $O(n*log)$，空间复杂度为 $O(n)$。

来源

CF 249D

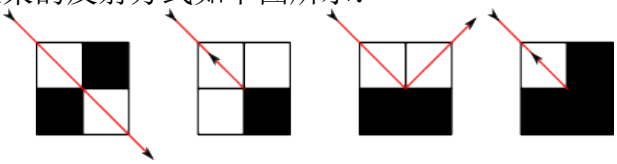
99 Mirror Room

题目大意

有一个 $n*m$ 的网格，左上角格子是 $(1,1)$，右下角格子是 (n,m)。

有 k 个堵塞的格子，其他格子是空的。

现在在 (xs,ys) 的中心向一个对角线方向（东北，西北，东南，西南）发射一束激光。如果光束碰到堵塞的格子或网格边界会反射。在不同情况下光束的反射方式如下图所示：



过了一会儿，光束进入了一个无限的循环。计算至少被光束通过一次的空格子数。

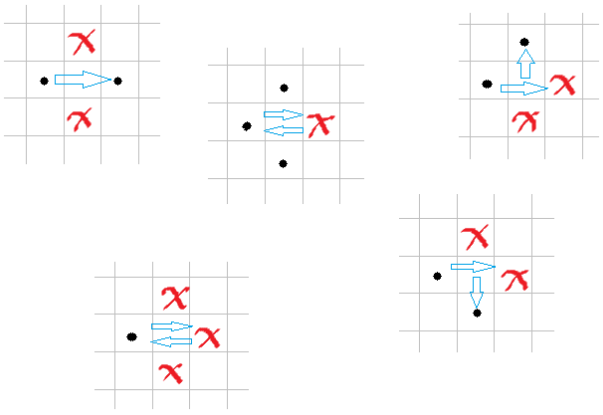
我们认为光束通过了一个格子的中心才算是通过了这个格子。

数据范围：$1 \le n,m \le 10^5, 0 \le k \le 10^5$

题解

我们把坐标轴逆时针旋转 45 度。

光束行走情况便图下图所示：



可以发现的是如果把所有转弯角都离散出来，那么线段条数是 k 级别的。

最后的情况可能是链或整个环。

那么我们只需要先走 k 步，如果走到链的一段则停止，否则就一定处在环中。

那么现在我们再模拟继续走，走到一个点出现两次就退出。

现在我们拥有一些平行于坐标轴的线段，这些线段会有交，但方向相同的线段不会有交，不同的方向的话最多交一个格子。

那么我们就可以用扫描线加树状数组，然后由于转坐标后有一半的点是无用的，那么维护的时候分开考虑就好了。

时间复杂度为 $O(n*log)$，空间复杂度为 $O(n)$。

来源

CF 274E

100 Cow Schul

题目大意

有 n 次考试，第 i 次考试得分为 Ti，而满分为 Pi。

老师在计算总成绩 G 前，把分数率 Fi 最低的 d 份试卷去掉，其中 $Fi = \frac{Ti}{Pi}$。

然后计算剩余 Ti 之和以及剩余 Pi 之和，总成绩 $G = \frac{\sum Ti}{\sum Pi}$。

求所有满足以下条件的 d：

去掉 d 份试卷，但总成绩 G 可以比老师算出来的更高。

没有两次考试分数率是一样的。

数据范围：$1 \leq n \leq 50000, 0 \leq Ti \leq Pi \leq 40000$。

题解

对于此题，我们可以设 Hi 为去掉分数率最小的 i 次考试后的总成绩。

然后我们需要验证这个 Hi 是否是去掉 i 次考试中总成绩最优的。

对于一个 Hi，对于任意一个剩余的考试集合 S，我们需要检验是否存在一个 S，使得 $\frac{\sum Tj}{\sum Pj}, j\epsilon\ S > Hi$。

那么可以把式子转化成 $\sum(Tj - Pj * Hi), j\epsilon\ S > 0$。

那么我们就可以得出了 $O(n^2)$ 的做法：

对于一个 Hi，我们在前 i 小的分数率的分数中算出最大的 $(Tj - Pj * Hi)$，在后 $n - i$ 的分数中算出最小的 $(Tj - Pj * Hi)$。如果前者的最大比后者的最小大的话说明把前 i 小的分数率删掉并不是最优的方法。

现在我们来优化这个算法：

我们把每个考试的情况映射到二维平面上，每个考试为一个点：(Pi, Ti)。

对于一个 Hi，我们可以把 Hi 看作是一个斜率，然后把所有点对于斜率 Hi 的截距算出来，前 $n - i$ 大的截距的点就是最优情况的方案。

根据上个 $O(n^2)$ 算法，我们就需要在前 i 小的分数率的点中找截距最大的点和后 $n - i$ 个点中找截距最小的点。

截距是一条斜率为 Hi 的线去切这些点，那么我们可以对于前 i 小的分数率的点维护一个上凸壳，对于后 $n-i$ 个点维护一个下凸壳，运用二分就快速算出斜率 Hi 切这个凸壳得到的极值了。

我们令 i 递增枚举，这样我们就是动态加一个点维护上凸壳和每个时刻询问用一个斜率去切这个凸壳得到的极值。那么这就变成了经典问题了。我们可以使用 cdq 分治或平衡树动态维护凸壳来解决，这两个算法都可以在 $O(nlogn)$ 的时间内解决。

而下凸壳只需倒着做就一样了。

如此此题就被解决了，**时间复杂度为 $O(n*log)$，空间复杂度为 $O(n)$**。

来源

USACO JAN 07 GOLD