IOI2016 国家集训队作业

试题泛做

佛山石中龙耀为

目 录

常规题（88）

Challenge 题（10）

1、Knight Moving

1.1 题意

有一张无限大的方格棋盘，上面有一个骑士。有 K 个格子是障碍格，骑士不能进入。我们的任务是将骑士从(0,0)移动到(n,m)。移动规则如下：每一步骑士能从(u,v)移动到(u+A,v+B)或(u+C,v+D)。求方案数。

1.2 数据范围

多测，$1 \le T \le 5$
$0 \le K \le 15$

1.3 关键字 容斥原理 分类讨论

1.4 题解

先考虑这个问题的简化版：把障碍格子去掉。此时可以根据题意得出以下二元一次方程，设使用规则 1 移动 s 步，使用规则 2 移动 t 步：

$$\begin{cases} As + Ct = n \\ Bs + Dt = m \end{cases}$$

除去方程无解或者多解的情况（这些情况可以轻易解决），此时 s 和 t 有唯一解。

那么 C_{s+t}^{s} 就是从(0,0)到(n,m)的方案数。

重新考虑原问题，我们可以通过容斥原理使得问题转化为：从(0,0)点出发，经过障碍格 k1,k2,k3……（步数即确定了次序），到达(n,m)的方案数。

1.5 复杂度

时间复杂度: $O(2^K \cdot K)$

2、Annual Parade

2.1 题意

有一个 N 个点，M 条边的有向图，每条边都有边权。有一些英雄，每位英雄的行走路径从 u 出发经过一些点到达 v。u 可能等于 v，但英雄至少要经过一条边。

费用包含三个部分：

1、英雄经过的边的边权和（按每人次计算）；

2、路径起点和终点不同的英雄每人付出 c 的费用；

3、没有被英雄经过的点每个付出 c 的费用；

询问 K 次，每次对于 c，计算最小花费。

2.2 数据范围

$2 \le N \le 250$
$1 \le M \le 30000$
$1 \le K \le 10000$

2.3 关键字 费用流

2.4 题解

先考虑问题的简化版本：c 确定的情况。

可以考虑每个点被哪条边到达（如果有多条则任选一条），称该点被这条边覆盖。如果一个点没有被任何一条边覆盖，那么只有两种情况：英雄没有经过这个点或是这是一个英雄的起点，此时要付出费用 c。

此时构图就水到渠成了：

每个点拆成入点和出点。

1、源点与出点相连，入点与汇点相连。

2、对于原图中的一条边(u,v)，u 的出点连向 v 的入点，费用即为该边边权。这条边流过意味着 v 由原图的边(u,v)覆盖。

3、对于原图中的一条边(u,v)，u 的出点连向 v 的出点，费用即为该边边权，流量无穷。意味着经过（或覆盖）u 点的英雄经过了这条边，但没有覆盖 v 点。

4、源点向入点连边，费用为 c，即该点没有被英雄经过或是一个英雄的起点。

做一遍费用流就 ok。

回到原问题，我们可以注意到只有第四类边与 c 相关。我们可以枚举第四类边有 i 条（加一个点限制）时的答案记为 $Ans[i]$，对于 c 答案即为 $\max\{Ans[i]+c(n-i)\}$。

更加巧妙地，我们可以将第四类边去掉。费用流流量 i 时的最小费用即为 $Ans[i]$。

2.5 复杂度

时间复杂度 $O(flow(n,m)+kn)$

可以通过 floyd 求出最短路径图优化第三类边的边数

3、 *Two Companies*

3.1 题意

有一棵 N 个点树，有两个树上的路径的集合 A 和 B，分别包含 M1 和 M2 条路径。集合中每条路径都有权值。任务是求 A 和 B 的子集 C 和 D，使得 C 和 D 中的路径不相交，并使得 C 和 D 中路径权值和最大。

3.2 数据范围

$1 \le N \le 10^5$

$1 \le M1, M2 \le 700$

3.2 关键字 最小割 LCA

3.3 题解

经典最小割模型，每个点代表一条路径。A 集合的路径连向源点，B 集合的连向汇点。如果两条分别属于 A 和 B 的路径有交，即在两点中连边，流量无穷。最后做一遍最小割，即最大流。

3.4 复杂度

时间复杂度 $O(flow(n, n^2))$

4、 *Sereja and Arcs*

4.1 题意

从左到右依次有 n 个点，每个点有一个颜色，第 i 个点的颜色为 c[i]。一个区间[u,v]存在当且仅当 c[u]=c[v]，且记该区间的颜色为 c[u]。询问有多少对不同颜色的区间相交了。

4.2 数据范围

$1 \le n \le 10^5$

4.3 关键字 分类讨论 树状数组

4.4 题解

先来考虑一个简单的问题：如何计算颜色 t 和其他颜色的区间相交的对数？

一个初步的想法：

弄出一个关于 t 的前缀和，以便统计区间内颜色 t 的点数。

枚举其他颜色 i，不妨先统计颜色为 i 的区间在右边，颜色 t 的区间在左边的相交对数（反过来同理）。称颜色为 i 的区间为 A 类区间，颜色为 t 的区间为 B 类区间。

从左到右依次枚举颜色为 i 的点 j，记 Ans[j]为[x,j]的 A 类区间与 B 类区间区间相交对数。设前一个颜色为 i 的点为 k，那么 Ans[j]比 Ans[k]多出来的部分即为：[1..k,k..j]的 B 类区间与[x,j]的 A 类区间的相交对数。这个问题通过计算贡献并不难解决。

如此我们便解决了颜色 t 与其他颜色区间相交对数的问题，复杂度为 $O(n)$。

通过这个方法我们可以先解决点数超过 $\sqrt{n}$ 的颜色。

对于点数小于 $\sqrt{n}$ 的颜色，我们可以注意到它们之间的区间数量为 $n\sqrt{n}$ ，通过树状数组可以在 $O(n\sqrt{n}\log_2^n)$ 的时间解决。

4.5 复杂度

时间复杂度 $O(n\sqrt{n}+n\sqrt{n}\log_2^n)$

通过调参可以优化到 $O(n\sqrt{n\log_2^n})$

5、Card Shuffle

5.1 题意

有 n 张牌叠成一叠放置，从上到下编号依次为 1 到 n，重复 m 次以下步骤：从牌堆顶拿走第一叠 A 张牌，再拿走第二叠 B 张牌，将第一叠牌放回牌堆顶，再拿走第三叠 C 张牌，将第二叠牌顺序反过来放回牌堆顶，最后将第三叠牌放回牌堆顶。

5.2 数据范围

$1 \le n, m \le 100000$

5.3 关键字 平衡树

5.4 题解

平衡树基本操作，可以通过 split 和 merge 的 treap 实现，将顺序翻转时打个标记即可

5.5 复杂度

时间复杂度 $O(m\log n)$

空间复杂度 $O(n)$

6、 *Misinterpretation 2*

6.1 题意

对于一个单词有以下一种重构的方法：将所有在偶数位置的字母依次拉到单词开头，然后剩下的字母依次排在后面。询问长度在[L,R]之间的单词（都是小写字母）有多少个重构之后与原单词相等。

6.2 数据范围

$1 \le T \le 5$

$1 \le L \le R \le 10^{10}$

$R - L \le 50000$

6.3 关键字 数论

6.4 题解

归纳题意可以得出长度为 n 的字符串满足下列特点：

当 n 为偶数，第 i 位字符与第 2i%(n+1) 位字符相同，用函数表示即为 $f(x) = 2x(\mathrm{mod}(n+1))$，设该置换循环为 $g(n)$。当 n 为奇数，$g(n) = g(n-1)+1$。

最终有 $Ans = \sum_{n=L}^{R} 26^{g(n)}$。

不妨只讨论 n 为偶数的情况，设 $ord(d)$ 为 2 模 d 的阶。可以证明 i 所在的循环长度为 $ord((n+1)/\gcd(i,n+1))$。所以 $g(n) = \sum_{p|(n+1) \cap p \ne 1} \varphi(p)/ord(p)$

将 n+1 分解质因数，然后通过 dfs 即可求出 n+1 的所有因子和欧拉函数值，我们可以发现 n 是连续的，所以只需要将 1 到 $\sqrt{R}$ 的质数分过去即可。

因为有 $ord(xy) = lcm(ord(x), ord(y))(\gcd(x,y)=1)$，所以求出所有质数幂的 ord 即可。设要求 $ord(t)$，则从 t-1 开始对每个质因子不断试除即可。

6.5 复杂度

时间复杂度 $O(\frac{R^{\frac{3}{4}}}{\log R} + \sqrt{R}\log^2 R + T(R-L)(\frac{\sqrt{R}}{\log R} + \log^2 R))$

7、 Sereja and Equeality

7.1 题意

两个长度为 n 的数组 A,B 相似，当且仅当对于所有 i(1≤i≤n)，满足 C(A,Ai)=C(B,Bi)。其中 C(X,x)等于满足 X[j]<x(1≤j≤n)的 j 的数目。

对于两个排列 P1,P2，定义函数 F(P1,P2) 等于满足 P1[l...r] 相似于 P2[l...r](1≤l≤r≤n)并且 P1[l...r]包含不超过 E 个逆序对的数对(l,r)的数目。

询问对 P1,P2 取遍所有 n 个元素的排列 F(P1,P2)的总和是多少。

7.2 数据范围

多测，$1 \le T \le 10000$

$1 \le n \le 500$

$1 \le E \le 1000000$

7.3 关键字 排列与组合 动态规划

7.4 题解

记长度为 $t \in [1,n]$ 的满足逆序对数量少于 E 的排列的数量为 $g[t][E]$，那么可得

$$Ans = \sum_{t \in [1,n]} g[t][E] \cdot C_n^t \cdot ((n-t)!)^2 \cdot (n-t+1)$$

至于如何求 $g[t][E]$，则为动态规划经典问题，在此不再赘述。

7.5 复杂度

时间复杂度 $O(n^3 + Tn)$

8、 Game of Numbers

8.1 题意

有两个长度为 n 的数组 A 和 B，还有两个集合 S1 和 S2，开始为空。

依次不断进行以下操作：取数对 (i,j) 和 (p,q)，满足 $(i,j) \notin S1$ 且 $(p,q) \notin S2$，$A_i < B_j$ 且 $A_p > B_q$，$\gcd(A_i,B_j) \ne 1$ 且 $\gcd(A_p,B_q) \ne 1$，且 $\gcd(A_i,B_j)$ 与 $\gcd(A_p,B_q)$ 不互质。然后将数对 (i,j) 和 (p,q) 分别加入 S1 和 S2 中。

询问最多能进行多少次操作。

8.2 数据范围

$1 \le n \le 400$

8.3 关键字 最大匹配 网络流

8.4 题解

对于所有合法数对 (u,v)（即 $\gcd(A_u,B_v) \ne 1$），将其分成两类：$A_u < B_v$ 为第一类，$A_u > B_v$ 为第二类。显然第一类只可能属于 S1，第二类只可能属于 S2。

将两类点做一下最大匹配即为问题的答案。

但直接做会超时，所以要优化一下连边。给有用的质数开一个点，对于 (u,v) 所代表的点，将 $\gcd(u,v)$ 分解质因数，将其与对应的质因子相连，此时边数从 n^4 降到 $9n^2$（最多有 9 个质因数）。

再将连边相同的点进行合并。

加上这些优化足以通过此题。

8.5 复杂度

时间复杂度 $O(n^2 \log n + flow(n^2, n^2 \log n))$

9、 *Observing the Tree*

9.1 题意

一棵 n 个节点的树，编号从 1 到 n，每个点点权初始为 0。要进行有 m 个操作，共有三类：

1. 将 x 到 y 路径上第一个点加 A，第二个点加 A+B，第三个点加 A+2B。
2. 求 x 到 y 路径上的点权和。
3. 将这棵树还原到 x 次操作后的状态。

9.2 数据范围

$1 \le n, m \le 10^5$

9.3 关键字 数据结构 可持久化 树链剖分

9.4 题解

树链剖分加可持久化线段树，接下来的基本操作不再赘述。

9.5 复杂度

时间复杂度$O(n\log^2 n)$

空间复杂度$O(n\log n)$

10、 Room Corner

10.1 题意

一间房子里的每个角落都有小孩。两个小孩可以沿着墙壁进行位置交换，移动一格时间为 1，更具体规则请看题目，太麻烦了就不写了。求一对小孩最短的相遇时间。

10.2 数据范围

$5 \le R \le 2500$
$3 \le C \le 2500$
即描述房间的图案的长和宽

10.3 关键字 模拟 二分查找

10.4 题解

先预处理出每对相邻小孩的距离，将该房间抽象成一个环。

每次询问一对小孩 u 和 v，则从顺时针和逆时针方向二分路径中点，取最佳答案。

10.5 复杂度

时间复杂度$O(RC+T\log(RC))$

11、 Fibonacci Numbers on Tree

11.1 题意

已知斐波那契数列由以下递推关系确定：$f(n)=\begin{cases} f(n-1)+f(n-2), n>2 \\ 1, 1\le n\le 2 \end{cases}$

有一棵 n 个节点的树，从 1 到 n 编号，需要处理 m 个操作。
共有四类操作：

1. 将 x 到 y 路径上的第 k 个节点权值增加$f(k)$。

2. 视 x 为根，询问以 y 为根的子树的节点的权值和。
3. 询问 x 到 y 路径上节点的权值和。

4.将整棵树返回到 x 次操作之后的状态。

11.2 数据范围

$1 \le n, m \le 10^5$

11.3 关键字 数据结构 可持久化 树链剖分

11.4 题解

对整棵树进行树链剖分，然后以 1 为根按 dfs 序弄出一个序列（重链优先），使得该序列中每条重链和每棵子树均为一个连续的区间。

考虑用可持久化线段树维护该序列，线段树每个节点记录该节点所代表的区间的前两项的权值，如此所有的权值都可以通过前两项递推得出。

第二类操作涉及到换根。经过观察，可以发现换根之后每个节点的子树要么不变，要么变成了补集，都可以轻易在线段树中查询。

11.5 复杂度

时间复杂度 $O(n\log^2 n)$

空间复杂度 $O(n\log n)$

12、 Rectangle Query

12.1 题意

给定一个二维笛卡尔平面，操作有 Q 条，需要支持下类三种操作：

1.插入一个左下角在（x1,y1),右上角在（x2, y2)的矩形。

2.删除第 i 条插入的的矩形。（从 1 标号）

3.询问目前与左下角在（x1,y1），右上角在（x2,y2)的矩形有至少一个公共点的矩形数量。

12.2 数据范围

$1 \le Q \le 10^5$

12.3 关键字 CDQ 分治 数据结构

12.4 题解

对操作进行 CDQ 分治。

考虑操作区间[L,R]，设其中点为 mid，计算区间[L,mid]内的修改对区间[mid+1,R]的询问的贡献。

注意到直接计算不太方便，可以考虑计算补集，即与询问区域不相交的矩形数量，通过排序加树状数组即可。

12.5 复杂度

时间复杂度$O(n\log^2 n)$

空间复杂度$O(n)$

13、 *Counting Hexagons*

13.1 题意

有一大堆长度在[1,N]的木棍，每种长度都有无数条，我们希望从中选六根使得能拼出一个面积为正的六边形，并且满足以下性质：

1.最长的那一根木棍的长度大于等于 L

2.其他木棍的长度都小于等于 X(X<L)

3.同样长度的木根不能超过 K 根

计算有多少不同的合法六边形。

13.2 数据范围

$2 \le N \le 10^9$

$2 \le L \le N$，$N - L \le 100$

$1 \le X < L$

$1 \le K \le 5$

13.3 关键字 数位 dp

13.4 题解

枚举最长的木棍的长度 LL，然后进行二进制数位 dp。

$f[i][j][k][P][Q]$表示处理到第 i 位，j 表示将五个数从小到大排序相邻两个的相等情况（用 2^4 记录下来），k 表示和大于 LL（k=0）或和大于 LL（k=1），P 表示最大值小于 X（P=0）或最大值等于 X（P=1），Q 表示进位。

进行一遍 dp 最后统计合法的情况即可。

13.5 复杂度

时间复杂度$O(2^4 \cdot 2 \cdot 2 \cdot 5\log N \cdot (N-L))$

空间复杂度$O(2^4 \cdot 2 \cdot 2 \cdot 5 \cdot \log N)$

14、Short

14.1 题意

给你两个数 n，k，你需要找出所有的数对(a,b)，满足 n<a<k,n<b<k，并且 ab-n 可以被(a-n)(b-n)整除。

14.2 数据范围

$0 \le n \le 10^5$，$k \le 10^{18}$

14.3 关键字 数论 分类讨论

14.4 题解

由题意可得

$$\frac{ab-n}{(a-n)(b-n)} = k, k \in N^*$$

经过简单的变形可得

$$b-n = \frac{n(a-1)}{k(a-n)-a}$$

$\because b-n > 0 \therefore k > 1$

不妨设 $a \le b$，得

$$a \le \frac{n(a-1)}{k(a-n)-a} + n$$

即 $a \le \frac{k+1}{k-1}n$，当 k=2 时 a 最大为 3n

我们可以设一个分界点 d。当 $a \le d$ 时暴力检查 $n(a-1)$ 的每一个因子是否能令 b 为正整数；当 $a > d$ 时我们可得 $k > \frac{a}{a-n}$，从小到大枚举 k 求出 b，如果此时 $b > a$ 则统计答案，否则退出循环。

14.5 复杂度

这个方法时间复杂度不好估计，调参调的好应该是$10^{6\sim7}$ 左右

15、 Fibonacci Number

15.1 题意

众所周知斐波那契数列定义为 $f(n)=\begin{cases} n, n\le 1 \\ f(n-1)+f(n-2), n>1 \end{cases}$

求最小的非负整数 n，满足 $f(n)\equiv C(\bmod P)$

P mod 10 是完全平方数

15.2 数据范围

$1\le T\le 100$

$11\le P\le 2\times 10^9$，P 为质数

$0\le C\le P-1$

15.3 关键字 数论 二次剩余 二次互反率 大步小步算法

15.4 题解

斐波那契数列的通项公式为

$$f(n)=\frac{(\frac{1+\sqrt{5}}{2})^n-(\frac{1-\sqrt{5}}{2})^n}{\sqrt{5}}$$

经过二次互反率可以证明 5 是 P 的二次剩余。

令 $x=\sqrt{5}$， $y=\frac{x+1}{2}$

可得 $y^n-(-\frac{1}{y})^n=xC$

设 $z=y^n$， $t=xC$

当 n 为奇数时，

$z+z^{-1}=t$

变形得 $z^2-zt+1\equiv 0(\bmod P)$

通过解一元二次方程（用二次剩余，有可能无解，唯一解或者两个解）得出根 m

即可得出 $y^n=m$，用大步小步求出 n 即可。

n 为奇数的情况类似。

15.5 复杂度

时间复杂度$O(\sqrt{P})$

16、 Chef and Balanced Strings

16.1 题意

一个平衡字符串被定义为其中的每个字符都出现了偶数次。
现在有一个字符串 P，由 n 个小写字母组成。
有 Q 次询问，三个参数 L，R，type。求

$$\sum_{\substack{L\le s<e\le R \\ P[s..e]\text{balanced}}} |P[s..e]|^{type}$$

强制在线。

16.2 数据范围

多测，$1 \le T \le 10^5$

N，Q 总和不超过 100000

16.3 关键字 分块

16.4 题解

先将问题进行转化：先将每个字母标个号（a 为 1，b 为 2，c 为 4……z 为 2^25）。

设函数 $g(x)=\begin{cases}0, x=0\\ g(x-1)xor(id[str[x]]), x>0\end{cases}$

P 的子串 P[L..R]为平衡字符串当且仅当 $g(L-1)=g(R)$

询问[L,R]，转化为询问[L-1,R]。

$$Ans=\sum_{L-1\le s<e\le R}(g(s)==g(e))\cdot(e-s)^{type}$$

考虑离线算法，发现用莫队算法能轻易解决这个问题。

假设当前询问区间为[L,R]，我们记录该区间的三个答案（长度 0 次，1 次，2 次和）。同时记录每种标号的个数，下标和，下标平方和（可以先离散化）。

即能用$O(1)$时间从[L,R]推到[L-1,R]，[L+1,R]，[L,R-1]，[L,R+1]。

如果强制在线的话，我们可以采用将莫队变成在线的常用手段分块。

$Ans[i][j](i,j\le\sqrt{n})$记录第 i 块到第 j 块的三个答案。

$sum[i][j](i \le n, j \le \sqrt{n})$ 记录第 i 种标号前 j 块的下标 0,1,2 次和。

16.5 复杂度

时间复杂度 $O(n\sqrt{n})$

空间复杂度 $O(n\sqrt{n})$

17、 Counting on a directed graph

17.1 题意

给定一个 N 个点（从 1 到 N 标号）M 条边的有向图。请你统计无序对(X,Y)的个数，其中(X,Y)满足存在一条从点 1 到点 X 的路径，和一条从点 1 到点 Y 的路径，且两条路径除了点 1 以外没有公共点。

17.2 数据范围

$1 \le N \le 10^5$，$0 \le M \le 5 \times 10^5$

17.3 关键字 图论 必经点

17.4 题解

不难发现这个问题的关键就是求每个点的最近必经点。求有向图的必经点有经典的必经点算法。

介绍这个算法之前，先来介绍一下半必经点：在搜索树 T 上点 y 的祖先中，不通过 y 到根的路径可以到达 y 的深度最小的祖先 x，称为 y 的半必经点。

求半必经点可以通过简单的并查集实现，这里不再赘述。

必经点定理：对于 G 中的一点 x，考虑搜索树 T 中 $semi(x)$ 到 x 的路径上除端点之外的点构成的集合。

设 $y = id[\min\{dfn[semi(z)] \mid z \in path\}]$

$$idom(x) = \begin{cases} semi(x), semi(x) = semi(y) \\ idom(y), semi(x) \ne semi(y) \end{cases}$$

通过该定理可以在求半必经点的时候顺便将必经点求出来。

17.5 复杂度

时间复杂度 $O(N+M)$，这里默认并查集复杂度为常数

18、 The Street

18.1 题意

有 n 个可重集合，每个集合有一个权值，一开始为 0。需要维护以下操作：

1.在第 L+i 个集合增加元素 b+i*a（0<=i<=R-L）

2.第 L+i 个集合权值增加 b+i*a（0<=i<=R-L）

3.询问第 i 个集合的最大元素与该集合的权值的和。

18.2 数据范围

$1 \le n \le 10^9$

$1 \le m \le 3 \times 10^5$

18.3 关键字 数据结构

18.4 题解

对于第二类操作可以通过简单的线段树实现。

问题的关键在于第一类操作，如果按照普通的线段树区间修改来实现，将会遇到标记无法合并的问题，以下有两种解决办法：

1、直接在线段树上每个节点上暴力套多一层数据结构维护这些一次函数的凸壳。

2、当要在其中一个节点打标记时发现原来就有标记时，可以发现这两个一次函数最多只有一个交点，所有新来的标记起码有一半是没用的，所以最多只需要往其中一边跑就可以了。

18.5 复杂度

时间复杂度 $O(m \log^2 m)$

空间复杂度 $O(m)$

19、 Chef and Graph Queries

19.1 题意

有一个无向图 G。顶点从 1 到 N 标号，边从 1 到 M 标号。

有 Q 对询问 $L_i, R_i (1 \le L_i, R_i \le M)$。对于每对询问，当仅保留编号 X 满足 $L_i \le X \le R_i$ 所在的边时候，图 G 中有多少连通块。

19.2 数据范围

多测，$1 \le T \le 1000$

N，M，Q 总和不超过 200000

19.3 关键字 数据结构 LCT 树状数组

19.4 题解

从小到大枚举右端点 R，$Ans[i]$ 表示询问 $[i, R]$ 的答案，考虑由右端点为 R-1 时的 Ans 更新得到右端点为 R 时的 Ans。

加入第 R 条边，如果没有形成环，那么 $Ans[1..R]$ 全部-1。

如果形成了环，那么查询环上最小编号的边记为 i，那么 $Ans[i+1..R]$ 全部-1。

用 LCT 维护森林即可。

用树状数组维护 Ans，将询问离线按右端点从小到大排序，在适当的时候记下答案即可。

19.5 复杂度

时间复杂度 $O(N \log N)$

空间复杂度 $O(N)$

20、 *Counting The Important Pairs*

20.1 题意

一张 n 个点 m 条边简单图（没有重边和自环），我们试图在上面删掉两条不同的边使得该图仍然联通，求不合法方案数。

20.2 数据范围

$1 \le n \le 10^5$

$1 \le m \le 3 \times 10^5$

20.3 关键字 hash 图论

20.4 题解

考虑该图的其中一棵生成树，将图上的边分成了两类：树边和非树边。

如果删除两条非树边，那么这肯定是一种合法方案。那么如何判断余下两种情况呢？

题解给出了一种方法：

将每条非树边标号，给每条树边赋予一个集合。如果一条树边被一条非树边覆盖，那么将该非树边的标号加进该集合。特别地，每条非树边的集合为只包含它自己标号的集合。

删除两条边 u 和 v 的是合法方案，取决于 u 和 v 的集合非空并且不相等。

证明应该是显而易见的。

进一步我们可以用 hash 来维护这个集合，先给每条非树边标一个 64 位整型的编号，集合内所有标号的异或代表该集合。

判断集合是否相等即异或值相等，集合为空即异或值为 0。

这种方法的正确率是可以接受的。

20.5 复杂度

时间复杂度 $O(m + m\log m)$（其中有排序的时间）

21、Counting D-sets

21.1 题意

在 N 维空间中的整点能组成多少不同点集，满足其直径恰好等于 D，点集的直径是在切比雪夫距离点集中最远的一对点的距离。

两个点集被认为是相等的，当且仅当它们可以通过平移相互得到。

21.2 数据范围

多测，$1 \le T \le 10$

$1 \le N \le 1000$

$1 \le D \le 10^9$

21.3 关键字 容斥原理

21.4 题解

考虑将问题转化为求直径小于等于 D 的不同点集数量，原问题可以通过 $Ans_D - Ans_{D-1}$ 得到。

我们称一个点集中一个维度是合法的，当且仅当该点集中有一个点该维坐标为 0。

不妨只计集每一维都合法的点集。其他点集通过给每个点该维的坐标不断减 1 会转化为这样的点集。

直接计算显得十分困难，但我们可以轻易计算最多有 i(0<=i<=N)个维度合法的点集，用容斥原理即可算出恰好有 N 个维度合法的点集。

21.5 复杂度

时间复杂度 $O(TN + N^2)$

22、A Game of Thrones

22.1 题意

1.一开始有 n 个数字写在一张纸上，第 i 种数字有 c[i]个。

2.A 和 B 轮流操作，A 先手。

3.第一轮中 A 先选择一个数字，把它称为当前数字。

4.之后从第二轮开始开始执行当前回合的人按如下操作：设当前数字为 u。将 u 从纸上擦去（如有多个随意擦一个）。选择另一个纸上的数字 v 满足与 u 刚好相差一个质因子称为当前数字。

5.无法完成操作的人输。

如两人都足够机智，谁将赢得游戏，如果 A 胜出还请输出第一轮的决策中能使他胜利的最小的数。

22.2 数据范围

$1 \le n \le 500$

$1 \le c[i] \le 10^9$，$1 \le u \le 10^{18}$

22.3 关键字 最大匹配 网络流 博弈论

22.4 题解

如果数字个数（即$\sum c[i]$）规模不大，则此题可以用最大匹配解决：

如果选了数字 u 下一个可以选数字 v（注意到这样的关系是双向的），则添加一条边(u,v)，这一步可以通过 Millar rabin 实现。

对这幅图进行最大匹配。

设起始点（即 A 一开始选的数字）为 k，若存在一种最大匹配 k 不是匹配点，那么 A 一开始选 k 必胜，否则必败。

证明如下：若当前的最大匹配 k 不是匹配点，A 先手选 k，接着 B 选择与 k 相连的数，从此刻开始，每轮 A 都选择 B 所选择的数的匹配点，由最大匹配的性质，最后肯定是 B 无法选择。若 k 必定是匹配点，与上述情况类似，B 每次选择 A 所选的数的匹配点，最后肯定是 A 无法选择。

但原问题中 c[i]规模太大，因此需要对上述算法进行改良。不难发现，该图只有偶环，即为二分图。我们将同一种数字用一个点来表示，将其与源点（汇点）连一条边流量为 c[i]。

进行最大流，可以发现一种最大流方案等价于一种最大匹配方案。若数字 k 不一定满流，A 先手选 k 必胜，否则 A 先手选 k 必败。

k 点是否满流用退流判断即可。

22.5 复杂度

时间复杂度$O(n^2 \log n + flow(n, n\log n))$

23、 *Two Magicians*

23.1 题意

一个 n 个点 m 条边的无向图，两位魔术师在进行一个游戏。

游戏开始前，第一个魔术师在 1 号点，第二个在 2 号点。

设当前回合的魔术师为 A。

第一步，A 可以沿着已有的路一直走，最终停在某一个房间，如果 B 也在该房间，A 胜利，否则继续。

第二步，A 必须增加一条原来没有的路。如果不能做到则失败，否则继续。

第三步，如果 A 的法力值为正数，他可以选择消耗 1 点法力瞬移到任何一个房间。

如果两个魔术师都足够机智，请问谁会取得最后的胜利？

23.2 数据范围

多测，$1 \le T \le 100$

$2 \le N \le 7777$，$0 \le M \le 10000$，$0 \le P \le 10000$

23.3 关键字 博弈论 dp

23.4 题解

为了叙述方便，我们同样称当前回合的魔术师为 A。

最暴力的 dp，即记录整个游戏局面：A 所在的联通块，B 所在的联通块，整幅图的联通情况，在不改变联通情况下可以增加的边数，双方剩余的法力值。

当然，这样的状态的数量堪称巨大。经过仔细推敲可以敏锐地发现：第四个状态（即不改变联通情况下可以增加的边数）可以进行优化。

设当前状态该值为 v。如果 v 为偶数，则等价于 v=0；若 v 为奇数，则等价于 v=1。

证明如下：我们可以将 v 值看成是可以进行 pass 操作的次数。首先有这样一个显而易见的事实，对于一个局面，剩余 pass 次数的改变不会影响对双方的有利程度。所以要么不进行 pass，要么 A，B 不停交替 pass 进行 v 次，就等价于进行 v%2 次，多出来的次数将毫无意义。

有了这个重大发现，那么状态就可以修改为：$f[i][j][k][l][p][q]$表示 A 目前所在联通块奇偶性为 i，B 所在联通块奇偶性为 j，含有奇数个点、偶数个点的联通块数量分别为 k 和 l，A 和 B 分别剩下 p、q 点法力值。

接下来题解给出若干神结论：游戏中两人总共最多使用一次传送，奇数块足够多（10 以上时）以 4 为周期循环，偶数块足够多时（10 以上）的时候 dp 值不变，最后状态数被简化为 $O(1)$ 。

23.5 复杂度

时间复杂度 $O(T)$

24、 Expected Maximum Matching

24.1 题意

按以下方式随机生成一个左边 n 个点右边 m 个点的二分图：左边第 i 个点和右边第 j 个点之间有边的概率为 f[i][j]。求这样生成的二分图的最大匹配的期望值。

24.2 数据范围

$1 \le n \le 5$

$1 \le m \le 100$

24.3 关键字 dp 最大匹配

24.4 题解

可以注意到左边的点数 n 不是一般的少，因此可以以右边的点为阶段进行 dp。

问题的关键就是如何表示状态，其实根据直觉来说状态数并不会太多，怎么表示没太大关系，方便就好。

这里采用题解的表示方法（题解的表示最方便）：n 个点共有(2^n)种匹配情况，直接开一个 32 位整数，每一位记录该种匹配情况是否可能出现就可以。

事实证明状态数数量级为 10^2。

24.5 复杂度

时间复杂度 $O(2^{2^n}m)$，事实证明远远达不到这个级别。

25、 Cool Numbers

25.1 题意

一个数有 k 位，每个数位上的数字分别为 X1, X2,..., XK，如果一个数 n 存在一到三个数位上的数的和为 s，且(X1 +X2 +... + XK - S)S 是 n 的倍数，那么 n 是 cool number。

定义 LC(N)和 RC(N)分别是小于等于 N 的最大的 cool number 和大于 N 的最小的 cool number，多次询问给定 N，求 LC(N)和 RC(N)。

25.2 数据范围

多测，$1 \le T \le 100000$

$1 \le n \le 10^{1000}$

n 的长度总和不超过 4000000

25.3 关键字 数论 分类讨论 高精度

25.4 题解

将 cool number 分为两类：
第一类为最多只有三位不为 0 的数，这一类很好处理。

第二类则为余下的数，设位数为 n，则有以下关系式 $(9n)^{27} \geq 10^n$。由此发现 n 不会太大，最多就 80。

所以只需枚举 cool number 的数位和（最多 720）i，然后暴力枚举 i^{27} 的所有因子进行判断就 ok（需要用到高精度）。
时限略紧可以通过打表打出一部分第二类 cool number，缩小枚举范围。
最后发现这类数大概有不到 40000 个，排个序，然后对于每个询问二分即可。

25.5 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。

26、 Ciel and Earthquake

26.1 题意

有一个(R-1)*(C-1)的网格图，即有 R*C 个路口，相邻的路口都有一条双向道路，发生地震时，每条道路有 p 的概率损坏，询问(1,1)与(R,C)联通的概率。

26.2 数据范围

多测，$1 \leq T \leq 50$

$1 \leq R \leq 7$，$1 \leq C \leq 10^{18}$

26.3 关键字 dp 数论

26.4 题解

R 的范围较小，可以采用连通性 dp 来解决这个问题，状态数大概 10^3 级别。
然而 C 的范围较大，若使用矩阵乘法状态数又过多，只能另辟蹊径。

通过看题解发现，当 R 固定，C 趋向无穷大时，$\dfrac{Ans(R,C+1,p)}{Ans(R,C,p)}$ 趋向于一个常数，所以 dp 做到 C' 为 40 到 50 时精度便足够。

26.5 复杂度

时间复杂度 $O(TSRC')$，其中 T 为数据组数，S 为状态数。

27、Evil Book

27.1 题意

有 N 个盒子，打开第 i 个盒子需要付出 C[i]的代价并获得 M[i]块钱。

在某一时刻，若此时你拥有至少 X 块钱，那么可以花费 X 块钱，使某一盒子降价，即该盒子的 C[i]和 M[i]都乘以 1/3。

一开始你一分钱都没有。

至少需要付出多少代价才能获得 666 块钱？

27.2 数据范围

多测，数据组数$1 \le T \le 5$

$1 \le N \le 10$

$10 \le X \le 666$

$0 \le C[i], M[i] \le 10^9$

27.3 关键字 搜索

27.4 题解

注意到，我们的目标很小，只需要 666 块钱。

假设目前我们在决策盒子 i。如果降价完之后 M[i]仍然大于 666，那么就已经达到了目的，这个状态的搜索终止，否则才有可能继续搜下去。注意到我们不可能给这个盒子降价花费比收益更多的钱。所以当降到 M[i]小于 666 时，这个盒子最多会被再降价两次，换句话说就是每个盒子最多有 3 种决策（降价不多于两次）。

直接用4^n作为状态，第 i 位为 0 则表示该盒子还未被打开，否则代表一种决策，搜索一遍判断每种状态的可到达情况，从可到达的状态里选一个最优的即可。

27.5 复杂度

时间复杂度$O(4^n \sum_{i=1}^{n} \log M[i])$

28、Dynamic Gcd

28.1 题意

一棵 n 个结点的树，每一个结点上有一个正整数权值，其中第 i 个结点上的权值是 v[i]，维护 2 种操作，共 Q 次：

1.找出从 u 到 v 的路径上所有点权值的最大公约数。（包括 u 和 v）

2.将从 u 到 v 的唯一路径上所有点权值加上 d。（包括 u 和 v）

28.2 数据范围

$1 \le n, Q \le 50000$

28.3 关键字 数据结构 树链剖分

28.4 题解

对点权进行差分（这对于 gcd 来说并不影响），将区间修改变为单点修改即可。

28.5 复杂度

时间复杂度 $O(n\log^2 n + Q\log^2 n)$

空间复杂度 $O(n)$

29、Equivalent Suffix Tries

29.1 题意

定义一个字符串的后缀字母树，即该字符串所有后缀所构成的字母树。

给定一个字符串，询问有多少个字符串的后缀字母树与该串的后缀字母树同构（只包含小写字母）。

29.2 数据范围

多测，数据组数 $1 \le T \le 10$

$1 \le n \le 10^5$，其中 n 为给定串的长度。

29.3 关键字 后缀数组

29.4 题解

不难发现该问题的关键是处理字符之间的相等关系。

由给定串的后缀树的叶子节点，我们可以确定从头开始若干后缀的相等关系，设为 $suffix[1..k]$。

$suffix[k+1]$ 肯定是前 k 个后缀其中一个的前缀（否则会有代表它的叶子节点），现在的问题就是决策选择哪一个后缀（换句话说，依附在哪个后缀上），设该后缀为 $suffix[j](1 \le j \le k)$。

由 $suffix[1..k]$ 之间的相等关系，$suffix[k+1]$ 的前面若干字母其实已经确定了，可以修改的就只有后面的字母。

同时还要维持 $suffix[1..k]$ 之间的相等关系，不要换成 $suffix[j]$ 之后使得 $suffix[1..k]$ 其中某对后缀能匹配更长的长度，等价于 $suffix[k]$ 的最长匹配不变。

上述两条即为合法的 $suffix[j]$ 的条件，直接枚举 j 进行检查就好，注意判重（具体实现可以有多种方法）。

29.5 复杂度

时间复杂度 $O(n\log n)$。当然，使用 DC3 能优化到 $O(n)$

30、Making Change

30.1 题意

询问有多少种方法可以凑出 C 块钱，假设每种硬币都有无限个。
两种方法不同当且仅当方案中存在一种硬币在两个方案中使用的次数不同。
不同种类硬币面值各不相同，而且任意两个硬币面值的最大公因数为 1。

30.2 数据范围

多测，$1 \le T \le 5$

$1 \le N \le 50$，$1 \le C \le 10^{100}$

$1 \le D_i \le 500$，面值

30.3 关键字 数学 生成函数 洛必达法则

30.4 题解

老实说，这题也是一知半解，膜完题解后对着公式和程序硬是敲了出来……结果最后还是不会做……

这里讲讲我会的部分：

首先写出生成函数

$$G(x)=\prod_{d\in D}\sum_{i}x^{id}$$

变化为

$$G(x)=\frac{1}{(1-x)^n}\prod_{d\in D}\frac{1}{\sum_{i=0}^{i<d}x^i}$$

然后通过 n 次单位根变成

$$G(x)=\frac{1}{(1-x)^n}\prod_{d\in D}\prod_{i=1}^{i<d}\frac{1}{x-W_d^i}$$

后面累乘那部分进行分式分解

$$\sum_{d\in D}\sum_{i=1}^{i<d}\frac{P_d(W_d^i)}{x-W_d^i}$$

通过洛必达法则可得

$$P_d(x)=\frac{(x(x-1))}{d\prod_{d'\in D}\sum_{i=0}^{d'\bmod d}x^i}$$

然后中间一串不知啥东东……
最后

$$Ans=\sum_{d\in D}\sum_{i=1}^{i<d}\frac{U_d(W_d^i)}{V_d(W_d^i)}$$

其中$U_d(x)=\dfrac{-(x^{(-S)\bmod d}+\sum_{j=0}^{j<n}C_{(S\bmod 10^9+7)+j}^{j}(1-x)^j)}{d}$

$$V_d(x)=\prod_{d'\in D\&d'\neq d}(1-x^{d'\bmod d})$$

求出$V_d(x)$关于$Q_d(x)$的逆元为$Z_d(x)$，其中

$$Q_d(x)=1+x+x^2+\ldots+x^{d-1}$$

再令$T_d(x)=Z_d(x)\times U_d(x)$

$$Ans=\sum_{d\in D}\sum_{i=1}^{i<d}T_d(W_d^i)$$

然后又弄来弄去搞出

$$\sum_{i=1}^{i<d}T_d(W_d^i)=d\times T_d(0)-T_d(1)$$

好了问题终于解决了……（真的是被出这道题的人打败了）

30.5 复杂度

时间复杂度$O(n^2D_i)$

31、Little Elephant and Colored Coins

31.1 题意

有 N 种类型的硬币，第 i 种硬币价值 V[i]，颜色为 C[i]，每种硬币数量无限。
询问恰好组成 S 美元的硬币最多能有多少种颜色？不合法输出-1。

31.2 数据范围

$1 \le N \le 30$

$1 \le V[i] \le 200000$，$1 \le C[i] \le 10^9$

$1 \le Q \le 200000$，其中 Q 表示询问次数，每次询问 $1 \le S \le 10^{18}$

31.3 关键字 dp

31.4 题解

初看此题有点难以入手。
不过我们不难想到一个十分暴力的 dp：

$f[i][j][k]$ 表示决策到第 i 种硬币，有 j 种颜色，目前恰好为 k 块钱是否可行。

或许我们能将第二维给去掉，但问题的关键明显不在这里，第三维的状态数实在是太多了。

有一个比较神奇的法子：选择一种硬币作为“基底”，不妨选择面值最小的那种，设其面值为 m。

我们发现对于所有需要组成的价值 $S \equiv S_0 (\bmod m)$，其中 S_0 为常数，有一种共性：

若 $S_0 + km$ 可以组成，那么 $S_0 + (k+1)m$，$S_0 + (k+2)m \cdots\cdots$ 都可以组成。

因此我们可以将第三维缩减到 m 的级别。

即 $g[i][j][k]$ 表示决策到第 i 种硬币，有 j 种颜色，最小的 S 是多少，其中 $S \bmod m = k$。

通过这个即可表示原本 $f[i][j][k]$ 包含的所有信息。

还有一个小问题，对于确定的 i 和 j，k 不具有特别明显的阶段，有可能出现依赖环。只需将依赖环从 dp 值最小的那里破开即可，因为大的值是无法更新小的值的。

31.5 复杂度

时间复杂度 $O(N^2 \min\{V[i]\})$

空间复杂度 $O(N \min\{V[i]\})$，此处使用了滚动数组优化。

32、A game on a graph

32.1 题意

两个人 A 和 B 在一个无向图 G 上进行游戏：
1.A 选择一个起始点，把一个硬币放在该点上。
2.接下来两个玩家轮流操作，B 先来。
3.每个玩家操作时，一定要把硬币沿着一条边移动到另一点。
4.硬币不能重复到达一个点。
5.无法操作的玩家将输掉游戏。
如果 A 和 B 都足够机智，询问有多少个点使得 A 选择该点为起始点能赢得游戏？

32.2 数据范围

多测，数据组数$1 \le T \le 3$
点数$1 \le N \le 2000$
边数$1 \le M \le 300000$
没有重边和自环

32.3 关键字 最大匹配 带花树算法

32.4 题解

最大匹配的经典模型，前文第 22 题已经有详细解释，这里不再赘述。

32.5 复杂度

时间复杂度$O(n^3)$

33、 Easy Exam

33.1 题意

设想有一个有 K 面的骰子，每个面上的数字分别是 1 到 K。同时给你两个参数 L 和 F(0<L<=K)。

将这个骰子掷 N 次。令 ai 为掷出数字 i 的次数。求 a1^F*a2^F*...*aL^F 的期望值（mod 2003 意义下）。

33.2 数据范围

多测，$1 \le T \le 2$

$0 < N, K \le 10^9$

$0 < L \cdot F \le 20000$
$0 < F \le 1000$

33.3 关键字 FFT 倍增 dp

33.4 题解

这题需要想清楚一个关键点，否则非常难以入手。

不妨先考虑 $F=1$ 的情况。

关键点就是将 a_i 拆开，即 $a_i=\sum_{j=1}^{N}x_{i,j}$ ，其中 $x_{i,j}$ 表示第 j 次是否掷出了数字 i，值为 0 或 1。

这一步将比较难处理的乘法变成了加法，通过计算贡献，即对于每种选择方案计算贡献，选择方案指的是对于每个 i 选取了哪个 j 令 $x_{i,j}$ 作为 1（即确定第 j 次掷出 i）。

$$Ans=\mathrm{P}_{\mathrm{N}}^{L}\cdot\frac{K^{N-L}}{K^{N}}$$

回到原问题，多了个指数 F 意味着，选择方案由原来每个 i 选择一个 j，变成了每个 i 选择 F 个 j。

当然，这 F 个 j 是可能有重复的，我们将这 F 个 j 去重后剩下的个数设为 T。再将从 T 个不同的 j 里选 F 次，当然每个 j 至少都要被选到一次的方案数设为 $g(T)$ 。

同样按照处理 $F=1$ 时的方法，我们只关注最后有多少次掷骰子被确定了，即 $\sum T_i$ 。

进行 dp， $f[i][j]$ 表示决策到了第 i 个数，共有 j 次掷骰子被确定了的方案数。

最后 $Ans=\sum_{i=F}^{L\cdot F}f[L][i]\cdot\frac{K^{N-i}}{K^{N}}$

现在问题就剩下求 f 和 g 了，它们的关系式不难推出：

$$f[i][j]=\sum_{T=1}^{f}f[i][j-T]\cdot C_{N-j+T}^{T}\cdot g(T)$$

我们最后只关注 $f[i][j]$ 所以可以将组合数分子部分提取出来，即

$$Ans=\sum_{i=F}^{L\cdot F}f[L][i]\cdot\frac{K^{N-i}}{K^{N}}\cdot\frac{N!}{(N-i)!}$$

$$f[i][j]=\sum_{T=1}^{F}f[i][j-T]\cdot\frac{g(T)}{T!}$$

令 $G(T)=\frac{g(T)}{T!}$

不难发现$G(T)$可以由以下 dp 推出

$$g[i][j] = j \cdot g[i-1][j] + g[i-1][j-1]$$

则$G(T) = g[F][T]$

用倍增和 FFT 优化求 f 的过程即可。

33.5 复杂度

时间复杂度$O(\log L \cdot n \log n)$，其中$n = N \bmod 2003$

34、 Music & Lyrics

34.1 题意

给出 n 个单词，给出 m 句歌词，求每个单词在所有歌词中出现的个数。

34.2 数据范围

$1 \le n \le 500$

$1 \le |P| \le 1000$，为每个单词的长度

$1 \le m \le 100$

$1 \le |S| \le 50000$，为每句歌词的长度

34.3 关键字 AC 自动机

34.4 题解

AC 自动机经典操作，不再赘述。

34.5 复杂度

时间复杂度$O(\sum |P| + \sum |S|)$

35、 Prime Distance On Tree

35.1 题意

给定一棵 n 个节点的树。如果我们在树中等概率地选取两个不同的点，这两个点之间的距离是一个质数的概率是多少呢？

35.2 数据范围

$1 \le n \le 50000$

35.3 关键字 FFT 树的分治

35.4 题解

对这棵树进行点分治。

对于一个重心，$sum[i]$ 表示过该重心的且距离为 i 的数量，这个用 FFT 不难算出来。

35.5 复杂度

时间复杂度 $O(n\log^2 n)$

36、 *Count Special Matrices*

36.1 题意

N(N>=3)是一个正整数。
A 是一个 N*N 的整数矩阵。
x 行第 y 个元素记作 A[x][y]
这个矩阵如果满足以下条件就称它是特殊的：

$A[x][x] = 0(1 \le x \le N)$

$A[x][y] = A[y][x] > 0(1 \le x < y \le N)$

$A[x][y] \le \max(A[x][z], A[z][y])(1 \le x, y, z \le N)$

$A[x][y] \in \{1, 2, \ldots N-2\}(1 \le x < y \le N)$

$\forall k \in \{1, 2, \ldots N-2\}, \exists x, y \in \{1, 2, \ldots N\} A[x][y] = k$

询问大小为 N 的特殊矩阵有多少个。

36.2 数据范围

多测，$1 \le T \le 10^5$

$3 \le N \le 10^7$

36.3 关键字 组合

36.4 题解

通过仔细观察条件三，我们可以发现：

对于 x,y,z（两两不等），如果确定了 $A[x][y]$ 和 $A[y][z]$ 且 $A[x][y] \neq A[y][z]$，那么 $A[x][z] = \max(A[x][y], A[y][z])$。

我们构造一个 N 个点的完全图，将 $A[x][y]$ 作为边 (x, y) 的边权。

我们按边权从小到大加入边，当目前要加入边权为 k 的边，加入后该边联通了两个联通块 A 和 B。根据上述发现不难得出 $\forall x \in A, y \in B, (x, y) = k$。所以在任意时刻，每个联通块都是一个完全图。

因此每次加边就只能连通两个联通块。所以只能加 N-1 次边。由于边权为 1 到 N-2 这意味着有且只有一种边权加了两次边。

加两次边有两种情况：将三个联通块连通起来，或将两对联通块连通起来（共有 3 种方法）

最后 $Ans = \sum_{k=3}^{N} \left(\frac{C_k^3 \cdot (3C_k^4)}{C_k^2 \cdot C_{k-1}^2} \prod_{i=2}^{N} C_i^2 \right)$

36.5 复杂度

时间复杂度 $O(N)$

37、Two k-Convex Polygons

37.1 题意

给定 n 个棍子的长度和整数 k，求能否在其中选出 2k 个棍子拼成两个凸多边形。使得两个凸多边形都恰好有 k 跟棍子组成,且任意相邻的边都不共线。

37.2 数据范围

$2k \le n \le 1000$
$3 \le k \le 10$

$1 \le length \le 10^9$，棍子长度

37.3 关键字 枚举

37.4 题解

先把棍子按长度从小到大排序，枚举两个凸多边形的最长的棍子第 i 根和第 j 根。

如果不冲突的话，那对于两个多边形肯定分别选择棍子 i-k+1 到棍子 i，棍子 j-k+1 到棍子 j。

冲突的话对于冲突部分直接搜索每条棍子给谁即可。

37.5 复杂度

时间复杂度 $O(knC_{2k-2}^{k-1})$，事实上远远达不到这个值。

还有一点，如果要构造 n 根不能组成一个 k 凸边形，n 不会太大，因为棍子长度有限制。所以对于该题如果 n 比较大，直接随意将棍子分成数量相等的两堆（或相差 1），两个凸多边形分别做即可。

38、Luckdays

38.1 题意

定义数列 S 为：

$$S[i]=\begin{cases}A,i=1\\B,i=2\\X\cdot S[i-1]+Y\cdot S[i-2]+Z)\bmod P,i>2\end{cases}$$

对每个询问 L[i],R[i]，求出满足 L[i]<=k<=R[i]且 S[k]=C 的 k 的个数。

38.2 数据范围

多测 $1\le T\le 2$

$2\le P\le 10007$，P 是质数

$0\le A,B,X,Y,Z,C<P$

$1\le Q\le 20000$，询问次数

$1\le L[i]\le R[i]\le 10^{18}$

38.3 关键字 大步小步算法 hash

38.4 题解

不妨先假设 $X>0$，$Y>0$。（等于 0 的情况可以特判掉）

定义二元组序列 $G(i)=(S[i],S[i+1])$，该序列肯定有循环节，且长度不超过 P^2。

而且因为 P 是质数，所以 $G(1)$ 肯定在循环节里面。

用大步小步算法求出循环节长度，经典算法不再赘述。
如何求出一个循环节里面有多少个 C 呢？

显而易见的，一个循环节里面最多只有一个 $(C,0),(C,1),(C,2)...$

对于每个 $(C,i)(0\le i<P)$，用大步小步算法判断是否在循环节里面即可，注意这里大步设为 $P\sqrt{P}$。

38.5 复杂度

时间复杂度 $O(P\sqrt{P})$

39、 *Colored Domino Tilings and Cutsontest*

39.1 题意

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，而且这两半都是合法的棋盘覆盖染色。

给出 N 和 M。构造一个棋盘覆盖染色使得，棋盘的割的数量最少。如果有多个解，还要使得使用的颜色最少。任意输出一组解。

39.2 数据范围

多测 $1\le T\le 3000$

$1\le N,M\le 500$

$N\cdot M$ 的总和不超过 2000000

39.3 关键字 分类讨论 构造

39.4 题解

好吧，各种分类讨论……画画图，耐心花点时间……

39.5 复杂度

时间复杂度 $O(nm)$

40、Sine Partition Function

40.1 题解

定义函数：

$$f(M,N,X)=\sum_{k_1+k_2+...+k_M=N}\sin(k_1X)\sin(k_2X)...\sin(k_mX)$$

其中 $k_1,k_2,......,k_m$ 都是非负整数

给出 M,N,X，任务是计算出 $f(M,N,X)$

40.2 数据范围

多测，$1\le T\le 10$

$1\le M\le 30$

$1\le N\le 10^9$

$0\le X\le 6.28<2\pi$

40.3 关键字 矩阵乘法 递推

40.4 题解

由各种三角变换可得

$\sin(k+1)X=2\cos X\sin kX-\sin(k-1)X$

得 $f(M,N)=2\cos X\cdot f(M,N-1)-f(M,N-2)+\sin X\cdot f(M-1,N-1)$ ，$M\ge 2,N\ge 3$

对于 $N<3$ 直接计算。

对于 $M=1$ ，$f(1,N)=2\cos X\cdot f(1,N-1)-f(1,N-2)$

然后直接矩阵乘法即可。

40.5 复杂度

$O(m^3\log n)$

41、 The Baking Business

41.1 题意

维护两种操作：

1.插入一个七元组 A(B)C(D(E))FG，权值为 H，括号表示里面的部分是可选的。

2.询问满足 $A=A',(B=B'),C=C',(D=D',(E=E')),F=F',L\le G(G\le R)$ 的七元组的权值和。括号表示里面的限制是可选的。

41.2 数据范围

$1\le S\le 10^5$

$1\le A,A'\le 10,1\le B,B'\le 3,1\le C,C'\le 10,1\le D,D'\le 20$
$1\le E,E'\le 5,0\le F,F'\le 1,1\le G\le 100,1\le L\le R\le 100$

41.3 关键字 统计

41.4 题解

直接开数组统计每个关键字的权值和即可，询问时暴力询问。

41.5 复杂度

时间复杂度 $O(S(R-L))$

空间复杂度 $O(ABCDEFG)$

42、 Max Circumference

42.1 题意

给出一个三角形 ABC，和 N 个操作，第 i 个操作可以使 A 的坐标加上 (x_i,y_i)，每个操作最多只能使用一次，最大化三角形 ABC 的周长，

42.2 数据范围

$K\le N\le 500$

42.3 关键字 几何 数学

42.4 题解

先摆出一条神奇的定理：设最后使得三角形 ABC 周长最长的 A 点坐标为 (X,Y)，那么必定有一组 (u,v)，使得 $f(x,y)=ux+vy$ 的最大值为 $f(X,Y)$。

证明也很简单，设最后三角形的周长为 D，那么 A 点肯定是在以 BC 为焦点，D 为长轴长的椭圆上，取 A 点的切线作为 (u,v) 即可。

对于一组特定的 (u,v)，我们可以独立计算每个操作的贡献 $f(x_i,y_i)$，把这些贡献从大到小排序，选出前 K 大的，如果正数数量不足 K 个则选择全部正数，即可得到该 (u,v) 下的方案。

然而选择合适的 (u,v) 并不容易。

有一种神奇的方法：

考虑 $f(x_i,y_i)=f(x_j,y_j)$，能够解出两组 (u,v)，并且关于原点对称，分成了两个半平面。那么这两个函数值的关系就取决于我们选的 (u,v) 落到了哪一个半平面上。

将这些相等关系解出的 (u,v) 极角排序，每当我们选的 (u,v) 转过一个向量，则意味着有一对操作的 rank 要调转（而且必定是相邻的一对），查询的时候二分即可。

这题还有一个小问题就是精度，此题精度要求十分高，所以排序时要用叉积，并且在开平方根的时候也需要一点小技巧：

设 $sqrt(S)=I+D$，其中 I 为整数部分，D 为小数部分。

整数部分比较容易得到。

小数部分：

$$I^2+2ID+D^2=S$$

$$D=\frac{S-I^2}{2I+D}=\frac{S-I^2}{I+\sqrt{S}}$$

这里 $\sqrt{S}$ 直接调用函数计算即可。

42.5 复杂度

时间复杂度 $O(n^2\log n)$

43、Ranka

43.1 题意

两名玩家在玩围棋，棋盘大小 9*9，规则如下：

在每一步中，玩家必须在一个空点放置一个棋子或者选择放弃本轮。如果玩家放置了棋子，可能有下面的情况发生：

若棋子放置后至少有一个对手的联通块呈无气状态，就将对手呈无气状态的棋子提出盘外，即"提子"。

否则，如果新放置的棋子会导致至少一个我方的联通块成为无气状态，则不能放置该棋子。

无气状态指该联通块没有与空点相邻。

还有一种规则叫"禁止全局同形再现"，不能出现两种局面有相同玩家操作相同的棋盘局面。

求一场含 N 个合法行动的比赛。

43.2 数据范围

$1 \le N \le 10000$

43.3 关键字 构造

43.4 题解

由规则可以注意到存在"一子翻盘"的可能性，即白棋（黑棋）已经布满了整个棋盘除了一个空点，如果将黑棋（白棋）放置在这个空点，就能将对方的全部棋子移除。

利用这个性质可以轻易构造一种方案：先白棋操作，布满整个棋盘剩下一个点，黑棋翻盘后再布满整个棋盘剩下一个点，以此不断循环。

注意全局同形再现的判断。

43.5 复杂度

时间复杂度 $O(N)$

44、Xor Queries

44.1 题意

给定一个初始时为空的整数序列(元素由 1 开始标号)以及 M 个询问：

类型 0：在数组最后加入数字 x。

类型 1：在区间 L..R 中找到数字 y，最大化(x xor y),xor 表示按位异或。

类型 2：删除数组最后 k 个元素。

类型 3：在区间 L..R 中，统计小于等于 x 的元素个数。

类型 4：在区间 L..R 中，找到第 k 小的数(第 k 个顺序统计量)。

44.2 数据范围

$1 \le M \le 500000$
$1 \le x \le 500000$
所有操作合法

44.3 关键字 数据结构 可持久化

44.4 题解

开一棵 $2^{17}=131072$ 的可持久化线段树，对于每个插入的数字转化为二进制后，该位为 0 则向左儿子走，为 1 则向右儿子走。
最大化异或操作时按位处理即可。
注意 y 的范围可能很大。

44.5 复杂度

时间复杂度 $O(m\log x)$

空间复杂度 $O(m\log x)$

45、 Union on Tree

45.1 题意

有一棵树，边长为单位长度。每个询问给出 m 个警卫，每个警卫 j 在点 a_j 并保护与点 a_j 距离不超过 R_j 的点，求有多少个点受到保护。

45.2 数据范围

$1 \le n, Q \le 50000$，Q 为询问数量

询问中警卫的总和不超过 500000

45.3 关键字 树的分治 虚树 可持久化线段树

45.4 题解

每次询问，我们先对警卫的保护范围 R_j 进行松弛操作。建出虚树，对于虚树的每一条边上的点，因为进行了松弛操作，所以只可能是由边两端的警卫进行保护。
对于一条边(u,v)，v 为靠近根节点的那一端，能轻易求出一个中间节点 w，使得

$R_u - dist(u,w) = R_v - dist(w,v)$，注意奇偶等细节。

可以发现，此时这条边上 u 到 w 的节点包括原树上的细枝末叶归 u 管，$Father[w]$ 到 v 的节点包括原树上的细枝末叶归 v 管。

设 $f(u,dis)$ 为原树中与 u 距离不超过 dis 的节点数。

设 $g(u,dis)$ 为原树 u 的子树与 u 距离不超过 dis 的节点数。

则 $Ans(Father[w],v) = g(v,R_v) - g(w,R_v - dist(w,v))$

$$Ans(u,w) = f(u,R_u) - g(u,R_u) - f(w,R_u - dist(u,w)) + g(w,R_u - dist(u,w))$$

把虚树上的每条边都按照上面的方法算一遍即可。

至于求原树中离一个节点距离不超过 x 的节点数可以用点分治进行预处理，求原树的一棵子树离该子树的跟距离不超过 x 的节点数可以用可持久化线段树进行询问。

45.5 复杂度

时间复杂度 $O(n \log n)$

空间复杂度 $O(n \log n)$

46、Children Trips

46.1 题意

给定一棵树，边权为 1 或 2。

每次给定点 u 和点 v，并给出距离 d。按如下规则行走：一开始在点 u，每天从当前点出发向点 v 走不大于 d 的距离到达一个点，直至到达点 v。询问从点 u 走到点 v 至少需要多少天。

46.2 数据范围

$1 \le N, Q \le 100000$，Q 为询问数量

46.3 关键字 分块 分类讨论

46.4 题解

1.当 $d > \sqrt{n}$，

将 u 到 v 的路径分为两段：从 u 到 LCA 和从 v 到 LCA。

不难发现，方向相反对答案并没有太大影响。

以 u 到 LCA 为例，从 u 一天一天往上走到最接近但没到 LCA 的点，记为 u'，同理得出点 v'。那么 $Ans(u,v)=Ans(u,u')+Ans(u',v')+Ans(v,v')$。

$Ans(u',v')$ 不难解决。

考虑如何求 u' 和 $Ans(u,u')$（$Ans(v,v')$ 同理）

考虑离线处理。

定义 $path(x,y)$ 为 x 到 y 的路径，y 为 x 的祖先。

每个询问分出来的两段路径 $path(u,u'),path(v,v')$。将 $path(x,y)$ 用链表记在点 x 处。

对这棵树进行 dfs，记 $dis[i]$ 为点 i 到根的距离，当到达一个点 $node$ 时，维护数组 $f[i]=j\,|\,\min\{dis[k]\,|\,dis[k]\ge i\}=dis[j]$，现在 node 往上走一天所到达的点为 $f[dis[node]-d]$。询问 x 到 y 的答案只需暴力一天一天往上走即可。

2.当 $d<\sqrt{n}$，

同样将 $path(u,v)$ 分成 $path(u,u')$ 和 $path(v,v')$，总体思路与第一类相同。

考虑 $path(u,u')$，我们可以每次走 $\sqrt{n}$ 天，再一天一天往上走。

维护在距离为 d 时一个点向上走一天，走 sqrt(n)天能到达哪个点，需要借助距离为 d-1 时的结果。

46.5 复杂度

时间复杂度 $O(n\sqrt{n})$

47、Chef and Tree Game

47.1 题意

一棵有根树 G，其上有 N 个节点，树上的边要么是黑色，要么是白色。

A 和 B 在树上玩一个游戏，规则如下：

两名玩家轮流行动；A 每一次删除一条还在树上的黑边，B 每次删去一条还在树上的白边；当一条边被删去之后，所有与根节点不连通的边也会一起被删去；当一名玩家无法行动时，他就输了。

如果 A 和 B 都足够机智，请问 A 先手时谁会胜利，B 先手呢？

47.2 数据范围

多测，$1 \le T \le 1000$

$1 \le N \le 100000$

所有 N 的和不超过 100000

47.3 关键字 博弈论 hackenbush

47.4 题解

PS：这题完全看题解……好吧感谢吉司机的题解。

给种局面定义一种权值，正负表示对哪一方有利，绝对值的大小表示有利的程度。

下面给出一些简单局面的权值：

1.一个孤立点，明显权值为 0 嘛，大家都一样。

2.两个点只有一条红边相连，姑且定义为权值为+1。拓展一下一棵 n 个节点全部由红边相连的树权值为+n

3.将第二种类型的红边改为蓝边，权值取反。

更具体一点，该局面函数 f 可以用以下递推方式得到：

1.叶子的函数值为 0。

2.每个节点的函数值为所有孩子的贡献之和，每个孩子 i 的贡献值按以下方法计算：

1.如果连接红边，那么令 a 为 $f(i)+a>1$ 的最小正整数，贡献为 $\dfrac{f(i)+a}{2^{a-1}}$

2.如果连接蓝边，那么令 a 为 $f(i)-a<-1$ 的最小正整数，贡献为 $\dfrac{f(i)-a}{2^{a-1}}$

别问我为什么……证明请看正版题解……

如果根节点的函数值为 0，那么谁后手谁赢；如果是正数，谁先手都是 A 赢；如果是负数，谁先手都是 B 赢。

函数的计算请用高精度……用链表维护诸如 $I+\sum_{i=1}^{m} ki \cdot 2^{w+Ai}$ 的形式即可，两个高精度数加减用启发式合并。

47.5 复杂度

时间复杂度 $O(n\log n)$

48、To Queue or not to Queue

48.1 题意

有一个字符串 S。初始时这个串是空的。

有一些操作，每个操作是下面两种之一：

1. 在 S 的末尾插入一个字符。
2. 删除 S 的第一个字符。

输出在每次操作之后当前串 S 的本质不同子串的个数的和。

48.2 数据范围

$1 \le Q \le 10^6$，操作个数

48.3 关键字 后缀树 Ukkonen 后缀数组

48.4 题解

正统题解的方法是在 Ukkonen 构后缀树的算法处理该问题。不过我不太会……

这里介绍一种用后缀数组处理该问题的方法：

先离线将整个字符串弄出来做后缀数组。

然后将操作反过来，即变为每次在前面增加一个字符，在后面删去一个字符。

我们现在要做的就是维护当前区间的 sa 数组，同时维护答案。

一般求一个字符串的不同子串个数都如此计算：

$$Ans = \sum_{i=0}^{n-1} len(suffix(sa[i])) - height[i]$$

这里我们也用同样的计算方法。

但直接维护 sa 不太行，随着每次在后面删掉一个字符，后缀的 rank 不是不变的，例如 aaz 和 az，同时删掉最后一个字符 z 后它们之间的排名就发生改变了，这给我们维护 sa 增加了很大的难度。

仔细观察每次改变排名的一对后缀，修改后肯定是其中一个是另一个的子串。考虑将每次修改后被包含的后缀删去（这肯定不影响答案），这样维护 sa 将变得十分简单。

对于排名相邻的一对后缀，设其在删到第 k 个字符后会出现包含情况。给每个位置拉一条链表，将这对后缀插入到第 k 个位置。每次删除字符的时候查询该位置的链表，删掉被包含的后缀即可。

对于插入操作，假设插入到 rank 为 i 的位置，判断 rank 为 i-1 和 i+1 的后缀有没有被包含即可（明显只有这两个后缀有可能被包含）。

48.5 复杂度

时间复杂度 $O(n\log n)$

49、Two Roads

49.1 题意

平面上有 n 个点。构造两条直线，使每个点到最近直线的距离的平方和最小。

49.2 数据范围

$3 \le n \le 100$

49.3 关键字 几何 数学

49.4 题解

考虑一个简化版的问题：构造一条直线。这个问题比较好解决，设该直线斜率为 k。

则 $Ans = \dfrac{\sum (x_i + ky_i)^2 - \dfrac{(\sum x_i + ky_i)^2}{n}}{k^2+1}$

$Ans = \dfrac{Ak^2 + Bk + C}{k^2+1}$，其中

$A = \sum y_i{}^2 - \dfrac{(\sum y_i)^2}{n}$， $B = \sum 2x_i y_i - \dfrac{2(\sum x_i) \cdot \sum y_i}{n}$， $C = \sum x_i{}^2 - \dfrac{(\sum x_i)^2}{n}$

设 $f(k) = \dfrac{Ak^2 + Bk + C}{k^2+1} = A + \dfrac{Bk + C - A}{k^2+1}$

则 $f'(k) = \dfrac{B(k^2+1) - 2k \cdot (Bk + C - A)}{(k^2+1)^2}$

当 $f'(k) = 0$ 时， $f(k)$ 有可能取最值

此时 $k = \dfrac{A - C \pm \sqrt{(A-C)^2 + B^2}}{B}$

带入 f 取最小值即为 Ans

不难发现最后的 n 个点肯定被两条垂直的直线（最优解的两条直线的角平分线）切开，“一三象限”和“二四象限”的分别归一条直线管。所以我们可以枚举两个点作 x 轴，再枚举一个点作 y 轴，然后两条直线分别计算即可。

49.5 复杂度

时间复杂度 $O(n^3)$

50、Martial Arts

50.1 题意

我方与对方在比赛，每对都有 N 个人。我方的第 i 个人打对方第 j 个人比分为 $A[i][j]:B[i][j]$。

由我方来安排比赛，每一个选手都需要安排恰好一场比赛。

当然我方想最大化我方和对方的得分差。

设我方最终得分为 H，对方最终得分为 G。那么我方的目的就是最大化 H-G，其次最大化 H。

但对方十分狡猾，他们能删去其中一场比赛，对方的目的是最大化 G-H，其次最大化 G。

怎样安排才能达到我方的目的？

50.2 数据范围

$1 \le N \le 100$

$0 \le A[i][j], B[i][j] \le 10^{12}$

50.3 关键字 最大权匹配 KM

50.4 题解

按边权从小到大枚举对方要删掉哪条边（即哪场比赛）。

此时我们要做的就是加入这条边，强制匹配这条边，并求出最大权匹配。强制匹配只需将该边边权修改为∞即可。

然后再将边权改为真实值，为下一次作准备。

用 KM 算法可以在$O(N^2)$内修改一条边的边权。

50.5 复杂度

时间复杂度$O(N^4)$

51、Arithmetic Progressions

51.1 题意

给定 N 个整数组成的数列$A_1, A_2 \ldots A_N$

询问有多少三元组(i, j, k)满足$1 \le i < j < k \le N$并且$A_j - A_i = A_k - A_j$

51.2 数据范围

$3 \le N \le 100000$，$1 \le A_i \le 30000$

51.3 关键字 FFT 分块

51.4 题解

将整个数列分为 $\sqrt{N}$ 块。

考虑第 i 块，设区间为[L,R]。

我们先用 FFT 将数组 T 算出来，$T[i] = \sum_{j=1}^{i-1} sumL[j] \cdot sumR[i-j], i \le 2\max\{A_k\}$

其中 $sumL[i]$ 为区间[1,L-1]中数字 i 的数量，$sumR[i]$ 为区间[R+1,N]中数字 i 的数量。

再从 L 到 R 枚举三元组中间那个数，设为 j，答案即为 $T[2j]$ 再加上其他数的贡献。

51.5 复杂度

时间复杂度 $O(\sqrt{N} \cdot m\log m)$，其中 $m = \max(A_i)$

52、Simple Queries

52.1 题意

给定一个含 N 个正整数的数组 A。现有关于它的 Q 个询问，询问有以下五种类型：

1.令 S 为由下标范围从 l 到 r 的不同的元素构成的有序集合。你需要求出

$(\sum_{1 \le i < j < k \le |S|} S_i S_j S_k)(\bmod(10^9+7))$

2.将下标为 x 的元素赋值 y

3.将下标为 x 的元素从数组中删除

4.在下标为 z 的元素之后插入元素 y，若 z 等于 0，则在数组最前端插入

5.输出下标在 l 到 r 范围内的不同元素个数

数组下标从 1 开始。数据保证数组总是非空。

52.2 数据范围

$1 \le N, Q \le 10^5$

52.3 关键字 数据结构

52.4 题解

问题的关键是去重，可以采用以下方法：对于每个 i 我们构造这样的一个二元组 (j,i)，其中 $j=\max\{k \mid k<i, A[k]=A[i]\}$。

询问区间[L,R]中的不同元素就是符合 $1\le j<L, L\le i\le R$ 的二元组 (L,R)。

去重解决后直接树状数组维护第一位，套个线段树维护第二位即可。

52.5 复杂度

时间复杂度 $O(N\log^2 N)$

空间复杂度 $O(N\log^2 N)$

53、 *Dynamic Trees and Queries*

53.1 题意

给定一个有 N 个点的有根树，执行以下 M 个询问。
每个节点有一个权值，一开始节点的编号为 0 到 N-1，根节点为 0。
询问有以下 4 种类型。
1.给 A 加一个权值为 v 的儿子，编号是从未使用的编号（编号为非负整数）中最小的一个。
2.将以 A 为根的子树中所有节点的权值加上给定值。
3.将以 A 为根的子树中所有节点删除。注意删节点不释放编号。
4.询问以 A 为根的子树中所有节点的权值总和。

53.2 数据范围

$1\le N,M\le 50000$

53.3 关键字 数据结构

53.4 题解

将这棵树的节点按照 dfs 序排列，即可将各种操作变为区间操作。
涉及到插入删除，需要用到平衡树。

53.5 复杂度

时间复杂度 $O((n+m)\log(n+m))$

54、 Seraja and Subsegment Increasings

54.1 题意

有两个包含 n 个整数的序列（整数范围为 0 到 3）A 和 B。

在一次操作中，可选择一个区间[i, j]，将所有 $A_k = (A_k + 1) \bmod 4, i \le k \le j$ 。

将 A 变为 B 需要至少执行多少次操作。

54.2 数据范围

$1 \le n \le 100000$

54.3 关键字 贪心

54.4 题解

考虑序列 $C_i = (B_i - A_i + 4) \bmod 4$ ，代表着 A_i 至少加 C_i 次 1 才能到达 B_i 。

由于每次是对区间操作，考虑将序列 C 差分，即 $S_i = C_i - C_{i-1}$ ，其中 $C_0 = 0$

如果什么都不改变，即给每个 A_i 恰好加 C_i 次 1，那么明显

$$Ans = \sum_{i=1}^{n} \max(S_i, 0)$$

只需从前往后每次操作尽可能往后延伸即可。

现在问题是有些位置操作次数可能不止 C_i 次，或许是 $C_i + 4k$ 次。

我们可以以这样的角度看 S 数组，正数项为需要付出的代价，负数项为可以继续向后延伸的操作数量。

我们的目的是令正数项的和尽量小。

我们可以将 $S_i = S_i + 4, S_j = S_j - 4(i < j)$ ，意味着将 $C[i..j-1]$ 加 4。

我们只可能这种操作达到我们的目的，按照贪心的策略，只有 $S_i = -3, S_j = 3; S_i = -2, S_j = 3; S_i = -3, S_j = 2$ 这三种情况对答案有利。

并不存在进行一次对答案不利的操作，使得出现了一次对答案更加有利的操作的情况，即使存在也能将其转化为上述的三种情况。

所以目标变成了尽量多地进行以上三种操作，这个目标很轻易就能实现。

54.5 复杂度

时间复杂度 $O(n)$

55、Trial of Doom

55.1 题意

一个房间被划分成了 n*m 个格子，每个格子可以是蓝色或红色。目标是从（1,1）走到（n,m），并使得所有格子都是蓝色的。每次可以移动到八个相邻的格子上，离开一个格子时这个格子和它周围的四个格子会改变颜色。询问能否达到目标。

55.2 数据范围

多测，$1 \le T \le 50$

$1 \le n,m \le 10^9$， $\min(n,m) < 40$

$0 \le k \le \min(nm,10000)$，k 为初始红色格子数量。

55.3 关键字 高斯消元

55.4 题解

不妨设 $n \le m$ 。

需要注意一点：我们可以不改变任何格子的颜色而进行移动（例如下-上-下-左上），前提是 n>1。

1. 当 n>1 时，

问题变成了每次改变一个格子与其相邻格子的颜色，能否将全部格子变为蓝色。

有这样的一个初步的想法，从 m-1 列往回推。设现在处理第 j 列，如果 $A[i][j+1]=1$，则在 $A[i][j]$ 这里取反。

经过这样的操作之后，除了第一列，其他的列全部为 0。或许运气好的话第一列也为 0，那么我们的目的就达到了。

根据这个思路，我们可以单独计算每一个格子对第一列的影响，即如果该格子为 1，推到第一列会使哪些格子为 1。

格子(i,j)对第一列的影响记为 $p[i][j]$，注意这里 $p[i][j]$ 为一个 n 位的二进制。

将最后一列的操作作为变量，每一行列一条方程，判断是否有解即可。

m 可能会很大，然而实际上 p 数组是有循环节的，循环节长度大概为 10^4

2. 当 n=1 时

先不考虑往回走的情况，先直接走一遍，求出当前的房间状况。

每次往回走可以改变对两个格子进行操作，所以判断当前房间操作数量是否为偶数即可。

55.5 复杂度

时间复杂度 $O(Len \cdot n + n^3)$，其中 Len 为循环节长度。

56、Billboard

56.1 题意

从左到右有 N 个方格，需要在任意连续的 M 个方格里放入至少 K 个物品，在满足条件的情况下物品数量最少。求方案数。

56.2 数据范围

多测，$1 \le T \le 3000$

$1 \le K \le M \le 500$

$M \le N \le 10^9$

56.3 关键字 杨氏图表

56.4 题解

数据范围里面有十分奇怪的一项：40%数据 M 整除 N。

这给了我们十分重要的提示：M 个 M 个来处理。

1.当 M 整除 N 时：

先将每 M 个方格分成一段，物品数量最少的方案肯定是每一段有恰好 K 个物品，我们确实能构造出这样的方案（例如每段前 K 个格子放物品），而且物品数量达到了下限。

不难发现方案应该满足这样的规律：

设第 i 块的第 j 个物品放在该块的 p[i][j]的格子处。

则 $p[i+1][j] \le p[i][j], 1 \le i < \frac{N}{M}, 1 \le j \le K, 1 \le p[i][j] \le M$

将该方案数记为 $f(\frac{N}{M}, M, K)$

2.当 M 不整除 N 时，设 t=M%N

明显最后一块与最后的 M 个格子有重叠，那当然尽可能地将物品往重叠部分塞，即每一块的[M%N+1,M]这一部分。

（1）$M - t = K, Ans = 1$

（2）$M - t > K, Ans = f(\frac{N}{M}, M - t, K)$

（3）$M - t < K, Ans = f(\frac{N}{M}, M + 1, t)$

f 值可以利用杨氏图表的计算公式来计算。

56.5 复杂度

时间复杂度 $O(TM^2)$

57、Selling Tickets

57.1 题意

有一个 n 个点，m 条边的图。求最大的数 k，使得任意选择 k 条边，这 k 条边中每条边都能选择它的一个端点，而且每个点只能被选中一次。

57.2 数据范围

多测，$1 \le T \le 15$

$2 \le n \le 200$

$0 \le m \le 500$

57.3 关键字 分类讨论 搜索

57.4 题解

既然目标是找最大的数 k，使得任意 k 条边都合法，那么可以将问题转化为寻找最小的数 k，使得其中 k 条边不合法。

直观上只要出现了两个联通的环就不合法，具体分为三种情况：

1.两个点之间有三条不相交的路径。

2.一个点连接两个环，这两个环有且只有该点作为交点。

3.两个环不相交的环，有一条路径连接它们。

Bfs 即可。

57.5 复杂度

时间复杂度 $O(Tn^2m)$

58、Little Elephant and Boxes

58.1 题解

有 n 个盒子，每个盒子里有一些钱或钻石。打开第 i 个盒子有 pi/100 的概率获得 Vi 美元，有(1-pi/100)的概率获得一颗钻石。

有 m 个物品，第 j 个物品需要恰好花费 Cj 美元和 Dj 钻石。

小象足够机智使得他获得一定量的钱和钻石后会买尽量多的物品。

询问期望购买物品的个数？

58.2 数据范围

多测，$1 \le T \le 5$

$2 \le n \le 30$，$1 \le m \le 30$

$1 \le V_i, C_j \le 10^7$，$0 \le D_j \le 30$，$0 \le Pi \le 100$

58.3 关键字 搜索 dp

58.4 题解

令 $f[i][j][k]$ 表示前 i 个物品用 j 个钻石买了 k 个最少花了多少美元，这个通过 dp 可以轻易求出。

比较直接的算法是直接 2^n 枚举每个盒子的状态，求出每次购买物品的数量。

当然可以利用常用的优化手段分开两半，一边为 a 个，另一边为 b 个。

先将它们的状态全部列出来放在 A 数组和 B 数组里。

对于 A 数组的每个状态 i，设它的钱和钻石总数为 $(sumC_i, sumD_j)$。再枚举一个 T，此时钻石的总数就确定为 $sumD_i + T$，此时钱被分为 m 个区间，每个区间都可以获得不同数量的物品，在 B 数组中钻石总数为 T 的状态中二分即可。

58.5 复杂度

时间复杂度 $O(n^3 + 2^a \cdot b^2 \cdot m + 2^b \cdot b)$，这里 $a+b=n$，可以取 $a=n/3$。

59、A New Door

59.1 题意

有一个 W*H 的矩形区域。有 n 个操作，每次给出一个圆形区域，把矩形区域与圆形区域的相交部分上色。最后把所有涂色部分挖掉，求矩形区域内（不包括矩形边界）空缺部分的周长。

59.2 数据范围

多测 $1 \le T \le 1000$

$1 \le n \le 1000$，输入中 n 的总和不超过 5000

59.3 关键字 计算几何

59.4 题解

比较基础的计算几何，涉及到圆与园求交点，园与矩形求交点。

枚举每个园，弄出它被覆盖的弧，对这些弧进行排序求出没有被覆盖的周长即可。

59.5 时间复杂度

$O(n^2 \log n)$

60、 Cucumber Boy and Cucumber Girl

60.1 题意

已知B个$N*N$矩阵$Q_1,Q_2...Q_B$。

对于每对(a,b)，$1\le a<b\le B$，构造一个新的矩阵$C_{a,b}$满足：

$$C_{a,b}[i][j]=\sum_{k=1}^{N}Q_a[i][k]\times Q_b[j][k],1\le i,j\le N$$

对于所有$1\sim N$的排列$p=\{p_1,p_2,...,p_N\}$，定义一个新序列；

$$Tr(p)[i]=C_{a,b}[i][p_i],1\le i\le N$$

用$CNT(a,b)$表示使得$Tr(p)$至少有一个奇元素的排列p的个数，定义数对(a,b)是合法的当且仅当$CNT(a,b)$是奇数。

计算有多少个数对合法。

60.2 数据范围

$1\le N\le 60$
$2\le B\le 8000$

60.3 关键字 数论 矩阵 Cauchy-Binet

60.4 题解

当 N=1 时，问题比较好解决。
现在我们假设 N>1。

对于每个(a, b)，我们构造$D_{a,b}[i][j]=(C_{a,b}[i][j]+1)\bmod 2$，此时(a, b)是合法当且仅当$\det D_{a,b}\bmod 2=1$

我们现在需要使用两个矩阵的乘法来表示$D_{a,b}$，以便使用 Cauchy-Binet 公式。

令所有的$Q_c[i][N+1]=1$，有$D_{a,b}[i][j]=\sum_{k=1}^{N+1}Q_a[i][k]\times Q_b[j][k]\bmod 2$

利用 Cauchy-Binet 公式，定义$Q_{a,c}$为矩阵Q_a删掉第 c 列，可得

$$\det D_{a,b} \bmod 2 = \sum_{c=1}^{N+1} \det Q_{a,c} \times \det Q_{b,c} \bmod 2$$

最后我们需要求出对于所有的 c， $\det Q_{a,c} \bmod 2$ 。

当 Q_a 的秩小于 N，所有的 $\det Q_{a,c} \bmod 2 = 0$ 。当 Q_a 的秩等于 N 的时候，行最简型类是由单位矩阵中添加一列形成的。

假设添加的这一列是第 k 列，它的第 i 个元素为 G_i ，对于 $i \ge k$ ， $G_i = 0$ 。

可以得出结论：

$$\det Q_{a,c} \bmod 2 = G_c, 1 \le c < k$$

$$\det Q_{a,c} \bmod 2 = 1, c = k$$

$$\det Q_{a,c} \bmod 2 = 0, k < c \le N+1$$

60.5 复杂度

时间复杂度 $O(N^2 B + B^2)$

61、Shortest Circuit Evaluation

61.1 题意

给出一条布尔表达式，请问在可以合法任意调转顺序的情况下，采用布尔短路计算期望计算次数最小是多少。

61.2 数据范围

多测， $1 \le T \le 50$
$1 \le Len \le 30000$ ， Len 为每个表达式的长度

61.3 关键字 模拟

61.4 题解

每个子表达式记录为 true 的概率和最小期望计算次数，模拟一遍即可。
去除了 not 之后，一个括号内肯定全是 and 或 or，将子表达式排序即可。

61.5 复杂度

时间复杂度 $O(T \cdot Len)$

62、Something About Divisors

62.1 题意

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*X。

62.2 数据范围

多测，$1 \le T \le 40$

$B \le 10^{12}$， $X \le 60$

62.3 关键字 数论 容斥原理

62.4 题解

令 $K = \dfrac{NX}{D}$，显然 $K < X$ 。

注意到 X 的范围十分小，我们可以枚举 K 来进行计算。

因为有重复，所以对于每个 K 统计满足 $K \mid NX$ 但不存在 $J \mid NX(K < J < X)$ 的 N

的数量，记为 $Ans[K]$ 。

令 $C[K] = \dfrac{K}{\gcd(K, X)}$， $N = C[K] \cdot M$

此时 $M \le \dfrac{BK}{X \cdot C[K]}$ 记为 $Limit$

另外还要减去满足 $\begin{cases} K \mid NX \\ J \mid NX \\ K < J < X \end{cases}$ 的 N 的数量

即减去满足 $\begin{cases} D[J] \mid M \\ K < J < X \end{cases}$ 的 M 的数量，其中 $D[J] = \dfrac{J}{\gcd(J, C[K] \cdot X)}$

利用容斥原理可得

$$Ans[K] = \sum_{\{J1, J2, \ldots, Jt\} \subseteq \{K+1, K+2, \ldots, X-1\}} (-1)^t \left[\frac{Limit}{Lcm(D[J1], D[J2], .., D[Jt])}\right]$$

接下来就要对该算法进行优化：

1. 如果存在 $D[J1] \mid D[J2]$，则可舍去 $D[J2]$

2. （核心优化）将 lcm 相等的状态合并在一起，关键是要进行排序。

由于扩展一次 $lcm' = lcm \cdot \dfrac{D[J]}{\gcd(lcm, D[j])}$，而 $D[J]$ 的因子数不超过 10，将相同因子数的归成一类，同一类之中已经有序，然后将 10 类进行合并即可。

62.5 复杂度

时间复杂度 $O(X \cdot S)$，其中 S 为状态数，大概在 10^4 级别，另外还有常数 10。

63、Chefbook

63.1 题意

构造 $P_x, Q_x (1 \le x \le N)$。

令 $W_{i,j} = L_{x,y} + P_x - Q_y$

满足 M 个约束条件，使得 $S_{x,y} \le W_{i,j} \le T_{x,y}$。

目标是令 $\sum W_{x,y}$ 最大，求最大值和一组方案。

63.2 数据范围

多测，$1 \le T \le 10$

$1 \le N \le 100$

$0 \le P_x, Q_x \le 10^6$

63.3 关键字 线性规划 费用流 互补松弛原理 差分约束系统

63.4 题解

不难得出线性规划模型：

在 M 个数对 (x, y) 中，令 $i (1 \le i \le N)$ 作为 x 出现的次数为 C_i、作为 y 出现的次数为 $-C_{N+i}$。同时令 $P_{N+y} = Q_y$，得：

$$\text{最大化} \sum_{i=1}^{2N} P_i C_i$$

$$满足约束\begin{cases} P_x - P_{N+y} \le T_{x,y} - L_{x,y} \\ P_{N+y} - P_x \le L_{x,y} - S_{x,y} \\ P_x \ge 0 \end{cases}$$

不难发现对偶之后问题能用费用流解决。

原问题：$\begin{cases} \max C^T P \\ AP \le B, P \ge 0 \end{cases}$，对偶问题：$\begin{cases} \min B^T Y \\ A^T Y \ge C, Y \ge 0 \end{cases}$

而且还有个很神奇的发现：

对偶问题中 $A^T Y \ge C$ 等价于 $A^T Y = C$

接下来就应该处理方案的问题了：
有互补松弛定理，对于

原问题$\begin{cases} \max C^T P \\ AP + U = B, P \ge 0 \end{cases}$，对偶问题$\begin{cases} \min B^T Y \\ A^T Y - V = C, Y \ge 0 \end{cases}$

若 X, Y 分别为原问题和对偶问题的最优解，则其充分必要条件为

$U^T Y + V^T X = 0$

这时上面的发现就派上用场了，$V = 0$

所以某个 $Y_i \ne 0$，那么就有 $P_x - P_y = B_i$

将这些约束和原来的约束弄出差分约束系统，即可得到原问题的一组最优解。

63.5 复杂度

时间复杂度$O(flow(N,M) + SPFA(N,M))$

64、 Find a Subsequence

64.1 题意

有一个长度为 N 的数组 A 和一个字符串 B，B 为“12345”的排列。求一个 A 的子序列，该子序列中的元素互不相等，且相对大小关系与 B 一样。

64.2 数据范围

多测，$1 \le T \le 20$
$5 \le N \le 1000$

64.3 关键字 贪心

64.4 题解

题目的要求仅仅是找一组可行解。
枚举第二个元素和第四个元素的位置。
剩下三个位置的元素按照这样的方法处理：
1.找出一个有上界或者有下界的（不可能没有）。
2.有上界的在合法情况下尽量取最大，有下界的同理。

64.5 复杂度

时间复杂度$O(TN^2)$

65、Team Sigma and Fibonacci

65.1 题意

给出M和N，求：

$$(\sum 6xyz\cdot fibo[x]\cdot fibo[y]\cdot fibo[z]) \bmod M, x+y+z=N$$

65.2 数据范围

$0\le N\le 10^{18}$，$1\le M\le 10^5$

65.3 关键字 数论

65.4 题解

PS：摘自吉司机题解
通过种种奇技淫巧可以算出该值关于N的函数的生成函数为：

$f(x)=6g(x)x^3(x^2+1)^3$，其中

$$[x^n]g(x)=\frac{1}{3125}[25C_{n+5}^5(fib_{n+6}+2fib_{n+5})+150C_{n+4}^4 fib_{n+5}+$$
$$5C_7^2C_{n+3}^3(fib_{n+4}+2fib_{n+3})+5C_8^3C_{n+2}^2 fib_{n+3}+$$
$$C_9^4(n+1)(fib_{n+2}+2fib_{n+1})+C_{10}^5 fib_{n+1}]$$

65.5 复杂度

时间复杂度$O(T\log n)$

66、Push the Flow

66.1 题意

给出一个无向图，每条边有容量，每个点最多属于一个简单环。实现一个数据结构，高效地处理两种操作：

1.求 G 以 S 为源点，T 为汇点的最大流。

2.修改某条边的容量。

66.2 数据范围

$1 \le n \le 100000$

$1 \le m, q \le 200000$

66.3 关键字 数据结构

66.4 题解

将环缩点，然后树链剖分。

重链上代表一个环的点权先按走重链计算。

询问时走轻边再重新计算每个环的权值。

66.5 复杂度

时间复杂度 $O(n\log^2 n)$

67、Different Trips

67.1 题意

有一棵 n 个点的树，其中城市 1 是根结点。

假如两个城市的度数相同，则他们被认为是相似的。

一条路径被这样定义：先选择一个点 A，再从其到根节点的路径上选一个点 B，A 到 B 即为路径。

两条路径被认为是相似的当且仅当他们的长度相同且按顺序一一对应的城市都相似。

询问有多少种不相似的可行的路径。

67.2 数据范围

$1 \le n \le 100000$

67.3 关键字 后缀数组

67.4 题解

在树上做广义的后缀数组即可。

67.5 复杂度

时间复杂度 $O(n\log n)$

68、 Payton numbers

68.1 题意

定义三元组的乘法 $multiply((a_1,b_1,c_1),(a_2,b_2,c_2))$：

$$s=(a_1a_2+b_1b_2+c_1c_2)+(a_1b_2+b_1a_2)+(c_1+c_2)$$

$$t=floor(\frac{s}{2})+16(c_1+c_2)-c_1c_2$$

$$A=t-2(a_1b_2+b_1a_2)-(a_1c_2+c_1a_2)+33(a_1+a_2)+(b_1b_2-a_1a_2)$$

$$B=t-5(a_1b_2+b_1a_2)-(c_1b_2+b_1c_2)+33(b_1+b_2)+(2b_1b_2+4a_1a_2)$$

```
if (s is even) then return (A-540,B-540,24)
else return (A-533,B-533,11)
```

定义 zero：若 x*任何 y=0，则称 x 是 zero
定义单位元，若 x*任何 y=y，则称 x 是单位元
定义质数，若 x 不是 zero 且不能分解成两个非单位元的乘积，则称 x 是质数
给定一个三元组，问他是不是质数。

68.2 数据范围

多测，$1\le T\le 10000$
$c=11$ 或 24

$-10^7\le a,b\le 10^7$

68.3 关键字 数论

68.4 题解

令 ω 为满足方程 $\omega^2=\omega-3$ 的解，即 $\omega=\frac{1+\sqrt{-11}}{2}$。

对于每一个三元组 (a,b,c)，都有到域 $Z[\omega]$ 的映射：

$$\phi(a,b,c)=(33-2a-c)+(b-a)\omega$$

现在要求判断域 $Z[\omega]$ 下的数 $a+b\omega$ 是否为质数。

定义共轭 $(a+b\omega)'=(a+b-b\omega)$，令 $Nx=xx'$。

即可得到神奇的结论：
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。

用 Miller-rabin 即可……

68.5 复杂度

时间复杂度 log 级别，具体看如何实现……

69、 Devu and Locks

69.1 题意

求有多少 N 位十进制数是 p 的倍数且每位之和小于等于 M，允许前导 0，答案对 998244353 取模。

69.2 数据范围

$1\le N\le 10^9$，$1\le P\le 16$，$1\le M\le 15000$

69.3 关键字 dp FFT 倍增

69.4 题解

一个比较直接的 dp：$f[i][j][k]$ 表示决策到第 i 位，前 i 位%p=j，前 i 位的和为 k 的方案数。

不难发现数位的决策顺序其实对答案没有影响。

1.考虑将一个 a 位的数和一个 b 位的数合并：

设 a 位的数的方案数为 $A[i][j]$，b 位的数的方案数为 $B[i][j]$。

则 $C[i+(10^i \bmod p)\cdot j]+=A[i]\times B[j]$

利用倍增可得，时间复杂度为 $O(\log n\cdot p\cdot M\cdot(p+\log M))$ 的算法。

2.将 $10^k \bmod p$ 相同的数位放在一起考虑：

枚举当前决策模 p 为 i 的数位，考虑将这一类的答案并到总答案里面。

先算出$T[j]$表示这些数位和为 j 的方案数（可以使用倍增+FFT）。

令$G[i \cdot j \bmod p][j] += T[j]$

用上面的方法合并总答案和 G 即可。

时间复杂度为$O(\log n \cdot p \cdot M \log M + p^3 M + p^2 M \log M)$

69.5 复杂度

时间复杂度

方法 1：$O(\log n \cdot p \cdot M \cdot (p + \log M))$

方法 2：$O(\log n \cdot p \cdot M \log M + p^3 M + p^2 M \log M)$

70、Query on a tree VI

70.1 题意

给定一棵 n 个节点的树，每个节点有一个颜色(黑/白)，初始都为黑。

维护一种数据结构，支持下列操作：

0 u：询问有多少点与点 u 连通。两个点 u 和 v 是连通的，当且仅当 u 到 v 最短路径上的所有点(包括 u 和 v)颜色都相同。

1 u：切换点 u 的颜色(黑变白，白变黑)。

70.2 数据范围

$1 \le n, m \le 10^5$

70.3 关键字 数据结构

70.4 题解

不妨设 1 号点为根，对整棵树进行树链剖分。

考虑对每个点记录以该点为根的子树，并且除去重链连接的子树，该点所在的联通块大小。

询问时只需一路往根找到该联通块深度最小的节点，在该点所在的重链上查询即可。

修改的时候一路往根走，每走过一条轻边就修改该点的答案，注意有增加答案和减少答案两种情况。

70.5 复杂度

时间复杂度$O(n \log^2 n)$

71、 Petya and Sequence

71.1 题意

给出一个长度为 n 的序列 $A[0..n-1]$，问是否存在一个长度为 n 的序列 $B[0..n-1]$ 满足：

至少存在一个 $0 \le i \le n-1$ 满足 $B[i] \ne 0$

对于任意 $0 \le j \le n-1$ 满足 $\sum_{i=0}^{n-1} A[i] \cdot B[(i+j) \bmod n] = 0$

71.2 数据范围

多测，$1 \le T \le 100$
$1 \le n \le 30000$，总和不超过 150000

$-1000 \le A[i] \le 1000$

71.3 关键字 矩阵 数论

71.4 题解

将问题转化为判断循环矩阵 X 是否满秩，其中 $X_{i,j} = A_{i+j \bmod n}$。

有以下定理：令 $f(x) = \sum_{i=0}^{n-1} A_i x_i$，令 $\gcd(f(x), x^n - 1)$ 的次数为 d，那么这个矩阵的秩就是 n-d。现在问题转化为判断 $f(x)$ 与 $x^n - 1$ 的公因式的次数。

有神奇的分圆多项式 $\phi_i(x)$，有以下性质：1. $\phi_i(x)$ 两两不同；2. $\phi_i(x)$ 不可约；3. $x^n - 1 = \prod_{d|n} \phi_d(x)$。

于是，问题又转化为了判断是否存在 $\phi_d(x)$ 满足 $d \mid n$ 且 $\phi_d(x) \mid f(x)$。

同时利用下面这条定理：

$a \mid b \Leftrightarrow a * p_1 p_2 p_3 ... p_k \mid b * p_1^{a1} p_2^{a2} ... p_k^{ak}$

其中 $\gcd(a, p_1 p_2 .. p_k) = 1$，$a_i \ge 1 (1 \le i \le k)$

所以 $\phi_d(x) \mid f(x) \Leftrightarrow x^d - 1 \mid f(x) * \prod_{p|d} (x^{\frac{d}{p}} - 1)$，其中 p 为质数

然后再利用一些小技巧即可解决这个问题。

71.5 复杂度

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

72、 Counting on a Tree

72.1 题意

给定一个包含 N 个节点的有标号有边权无根树，点从一开始标号。

你的任务是计算有多少无序数对(S,T)，满足连接 S,T 两点的路径上，所有边权的最大公约数等于 1。当然，我们只计算 S≠T 的数对。

给定 Q 组询问，其中第 i 组询问形如 Ai,Ci，表示第 Ai 条边的权值修改为 Ci。对于每一组询问，输出改动后对应的答案。

72.2 数据范围

$1 \le N \le 10^5$， $0 \le Q \le 100$

$1 \le C \le 10^6$，边权

72.3 关键字 莫比乌斯函数 并查集

72.4 题解

有这样的一条众所周知的公式：$\sum_{d|n} \mu(d) = (n == 1)$

所以只需对每个 d 单独计算即可。

从 1 到 $\max(C)$ 枚举 d，将边权为 d 的倍数的边用并查集并起来，然后计算在相同联通块里的点对数量，设为 T，则 $Ans += \mu(d)T$ 。

修改次数很少，只需暴力修改即可。

72.5 复杂度

时间复杂度 $O(N\sqrt{C}+NQ)$

73、 Minesweeper Reversed

73.1 题意

我们来进行扫雷的逆过程，每次可以关闭一个含雷方块，或者关闭一个不含雷的方块。每当关闭一个不含雷的方块时，所有可能与这个方块一起打开的方块都会被关闭。求最少操作次数

73.2 数据范围

多测，$1 \le T \le 50$

$1 \le R, C \le 50$

73.3 关键字 最大匹配 带花树算法

73.4 题解

明显雷和四周没有数字 0 方块的需要一个个点来关闭。
然后剩下的就是由数字 0 方块组成的一个个联通块。
有些联通块有“相交”，点在相交处可以同时关闭。
不过可以发现那些“连接”两个联通块的点最多只会连接两个联通块。
我们要尽量多地进行每次能关闭两个联通块的操作。
用一般图最大匹配就可以了。

73.5 复杂度

时间复杂度 $O(n^3)$，n 为联通块个数。

74、Attack of the Clones

74.1 题意

我们称一个形为 $f: A \to B$ 的函数叫做布尔函数，其中 A 是所有长度为 n 且仅由 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。

我们称满足一些条件的布尔函数构成的集合称为 clone。
现在有四个特殊的 clone 如下：

Z 是 0-保留值函数集合：满足 $f(0,...,0) = 0$；

P 是 1-保留值函数集合：满足 $f(1,...,1) = 1$；

D 是自对偶函数集合：满足 $!f(x_1,...,x_n) = f(!x_1,...,!x_n)$

A 是仿射函数集合：满足 如果 $f(a_1,...,c,...a_n)=f(a_1,...,d,...,a_n)$ 则 $f(b_1,...,c,...b_n)=f(b_1,...,d,...,b_n)$ 的函数，在这里 c 和 d 都在某个位置 i，并且这个对于任意 i，$a_1,...,a_n,b_1,...,b_n,c,d$ 都应成立。

现在我们有兴趣知道在上述几种集合的组合中有多少个 n 项函数。

74.2 数据范围

$1 \le n,q \le 100$，q 为询问的次数

表达式长度不超过 100

74.3 关键字 括号匹配 数论

74.4 题解

可以发现 n 项函数可以被是否属于 A，D，P，Z 这 16 个状态进行划分，而且做到了无重复无遗漏。

直接计算这些状态的 n 项函数个数 $F[i](0 \le i < 16)$ 有点困难，考虑计算 $G[i]=\sum_{j\&i==i} F[j]$，即每个 0 表示肯定不在该集合，1 表示可以在该集合。

经过艰苦的手算可得：

$$\begin{aligned}
&G[0]=2^{2^n}, G[1]=G[2]=2^{2^n-1}, G[3]=2^{2^n-2};\\
&G[4]=2^{2^{n-1}}; G[5]=G[6]=G[7]=2^{2^{n-1}-1};\\
&G[8]=2^{n+1}; G[9]=G[10]=G[12]=2^n;\\
&G[11]=G[13]=G[14]=G[15]=2^{n-1};
\end{aligned}$$

剩下的事情只需要开一个 2^{16} 的状态记录这 16 个状态被包含了即可。

74.5 复杂度

时间复杂度 $O(n)$

75、Sereja and Order

75.1 题意

Sereja 有 N 个程序，每个程序都需要在两台电脑上分别运行。第 i 个程序需要在第一台电脑上运行 A[i]秒，在第二台电脑上运行 B[i]秒。一台电脑不能同时运行两个程序，一个程序也不能同时在两台电脑上运行。Sereja 需要用最少的时间完成所有程序在两台电脑上运行的任务，请你帮帮他。

75.2 数据范围

多测

$1 \le N \le 10000$，总和不超过 200000

75.3 关键字 贪心

75.4 题解

通过大量的实践，我们可以发现：

一般情况下 $Ans = \max(\sum A[i], \sum B[i])$。

而当 $\max\{A[i]+B[i]\} > \max(\sum A[i], \sum B[i])$ 时，$Ans = \max\{A[i]+B[i]\}$

至于方案，我们可以直观地感觉到，方案应该是比较容易构造的，只要随机然后判断是否合法即可。

75.5 复杂度

时间复杂度 $O(rp \cdot n)$

76、Chef and Churu

76.1 题意

有一个含 N 个数字的数组 A，元素标号 1 到 N，同时有 N 个函数，也标号 1 到 N。第 i 个函数会返回数组中标号 L[i]和 R[i]之间的元素的和。进行一下两种询问：

1 x y 将数组的第 x 个元素修改为 y。

2 m n 询问标号在 m 和 n 之间的函数的值的和。

76.2 数据范围

$1 \le N, Q \le 100000$

76.3 关键字 分块

76.4 题解

初看这题十分地麻烦，各种求和数不胜数……

在线直接暴力上数据结构没什么好办法……

考虑离线算法，我们采用对付这种问题的老手段：将修改和询问一起分块，然后对于每一块进行单独处理。

先将块前的修改解决，求出该块内每个询问的答案。

接下来的问题就是如何解决块内的修改对询问的影响。

考虑对函数也进行分块，每一块算出该块的函数对每个位置包含了多少次，于是每

次修改只需对每一块计算影响即可。

此时询问区间内整块的函数已经解决，剩余头和尾大概$\sqrt{n}$个函数。

我们希望能做到$O(\sqrt{n})$修改某个点的值，$O(1)$询问某个区间的值。

还是有点困难，将问题转化为前缀和，于是变成了$O(\sqrt{n})$修改某个区间的值，$O(1)$询问某个点的值，这个又用一个分块即可。

76.5 复杂度

时间复杂度$O(n\sqrt{n})$

77、Short2

77.1 题意

给定一个质数 p，问有多少对(a,b)(a>p,b>p)满足 ab 被(a-p)(b-p)整除。

77.2 数据范围

$1 \le p \le 10^{12}$

77.3 关键字 数学 分类讨论

77.4 题解

设$A=a-p, B=b-p$，则题目的可以转化为问有多少对$(A,B)(A>0,B>0)$满足$AB \mid p(A+B+p)$。

分三类情况讨论：

1. $p \mid A$且$p \mid B$，可以发现答案为 5，$(\frac{A}{p},\frac{B}{p})=(1,1),(1,2),(2,1),(2,3),(3,2)$。

2. P 不整除 A 且 p 不整除 B。

于是$AB \mid (A+B+p)$，可得$p+1 \ge (a-1)(b-1)$。

当$A=B$时，只有$(1,1)$；否则不妨设$A<B$，这时满足$A<w=1+\sqrt{p+1}$，$B=\frac{A+p}{Ak-1}$，令$d=ak-1$，我们要找的是满足以下条件的(A,d)：（1）p 不整除 A；（2）$d \mid (A+p)$；（3）$A \mid (d+1)$；（4）$B>A$。

分两种情况：

（1）对于所有的$d \le \sqrt{p+w}$，此时只有两个A可能满足要求

（2）对于所有$B \le d$即$B \le \sqrt{p+w}$，此时只有一个A可能满足要求。

3. $A \mid p$和$B \mid p$其中一条成立，此时第二种情况中的每一对(A,B)都能转换为$(A, \frac{p(A+p)}{B})$和$(\frac{p(B+p)}{A}, B)$。

77.5 复杂度

时间复杂度$O(\sqrt{p})$

78、Hypertrees

78.1 题意

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。

78.2 数据范围

$3 \le N \le 17$

78.3 关键字 搜索

78.4 题解

有这样的一个暴力算法：对于每一棵超树，将其拆成若干个点双联通分量。可以发现对于任意一个点双联通的超树，每一条边必定会连接一个叶子节点（除了点数为3的情况）。删掉那个点之后就得到了一个常规的点双联通图，原来超图的点数就是这个图的点数+边数。

于是可以暴力搜出点数为i的双联通超树的数量。
接着再暴力将这些点双联通超树拼接起来即可。
然后打表就ok啦。

78.5 复杂度

暴力时间复杂度无法估计

打表复杂度$O(1)$

79、Black-white Board Game

79.1 题意

有一个 N*N 的矩阵，第 i 列的第 L[i]至第 R[i]被涂成了黑色。
定义合法的 1 到 N 的排列为：对于任意的 i，都有(i,P[i])为黑色格子。
询问是奇数逆序对的排列多还是偶数逆序对的排列多？

79.2 数据范围

多测，$1 \le T \le 15$

$1 \le N \le 10^5$

79.3 关键字 左偏树

79.4 题解

我们可以给这些排列再加一个限制条件：

不存在$1 \le i < j \le N$，使得$L[i] \le P[j] \le R[i], L[j] \le P[i] \le R[j]$。因为这样的排列是可以奇偶相互抵消的。

我们尝试从左到右确定每一列选中的方块。
先来考虑一下第一列应该怎么做：

假设有 i 行满足$L[a_i]=1$，设$k = a_j \mid \max\{L[a_i]\} = L[a_j]$，那么我们选择的格子肯定是(1,k)。否则根据我们新增的限制条件第 k 行就无法再被选择。如果 k 有两个或者多于两个的选择，意味着不存在这样的排列。

此时根据新增的限制条件，对于所有的$a_i \mid L[a_i]=1$，都有$L[a_i]=L[k]+1$。

以此类推进行每一列的操作，可以考虑使用左偏树来寻找 k。
最后得出一个确定的排列，判断奇偶，或者这样的排列不存在，输出平局。

79.5 复杂度

时间复杂度$O(n\log n)$

80、Little Party

80.1 题意

有 m 个 n 位二进制数S_i，我们可以选择一些数对(a_i,b_i)（a_i,b_i为 n 位二进制数）对其进行覆盖。

需要满足以下条件：$a_i \& b_i = b_i$，对于所有的 j，不同的$(S_j \& a_i)$有2^k个，其中 k 为a_i中 0 的数量。

S_j能被(a_i, b_i)覆盖当且仅当$S_j \& a_i = b_i$。

求一组覆盖的方案，使得a_i中 1 的数量的总和最少。

80.2 数据范围

$1 \le T \le 120$
$1 \le m \le 1000$
$1 \le n \le 5$

80.3 关键字 搜索

80.4 题解

先将S_j去重，此时 m 不超过2^n。

直接枚举所有数对判断是否合法，并求出能覆盖哪些S_j。

直接搜索即可，需要加许多优化，或许可以部分记忆化。

80.5 复杂度

搜索的时间复杂度不好估计，但不会太大。

81、 Find a special connected block

81.1 题意

在一个 N*M 的矩阵中选择一个联通块，使其包含 K 种不同的颜色。每个格子还有选择的代价，在合法的前提下询问最小代价。

81.2 数据范围

$1 \le N, M \le 15$

$1 \le K \le 7$

81.3 关键字 随机化 斯坦纳树

81.4 题解

初看这题十分的难搞，至少包含 K 种不同的颜色是什么玩意……

题解给出了一个神奇的随机化算法：随机给每种颜色重新赋予 1 到 K 之间的每种颜色，然后就是一个斯坦纳树的经典问题：

令 $F[i][j][k]$ 表示格子(i,j)所在的联通块中覆盖了 k（k 为一个 K 位二进制数）状态颜色的联通块最小代价。

于是有这样的转移：

1.按边转移：$F[i][j][k]+V[x][y]\rightarrow F[x][y][k]$，其中 (x,y) 与 (i,j) 相邻。

2.按点转移：$F[i][j][x]+F[i][j][y \wedge x]-V[i][j]\rightarrow F[i][j][y]$。

初始状态：如果一个非障碍的格子 (i,j) 有颜色 $A_{i,j}$，则 $F[i][j][2^{A[i][j]-1}]=V[i][j]$ 否则 $F[i][j][0]=V[i][j]$。

嗯这样做的正确率是十分高的，将近 99%。

81.5 复杂度

时间复杂度 $O(TNM2^K)$，T 为随机的组数，可以取 500。

82、Substrings on a Tree

82.1 题解

有一棵 n 个节点的树，每个节点都有一个小写字母。一个该树子串被定义为：从一个节点出发走到它其中的一个子孙节点的路径上字符依次连接所成的字符串。

询问该有多少个不同的子串，和给出小写字母间大小关系后，第 k 小的子串。

82.2 数据范围

$1 \le N \le 250000$

$1 \le Q \le 50000$

最后的输出不超过 800KB

82.3 关键字 后缀自动机

82.4 题解

直接在树上进行广义的后缀自动机即可。

82.5 复杂度

时间复杂度 $O(n+26Out)$，其中 Out 为输出规模

83、 Graph Challenge

83.1 题意

有一个 N 个点，M 条边的有向图，点用 dfs 编号。

一个节点 u 被称为另一个节点 v 的 supreme vertex，当且仅当存在一条有向路 $u=a_0,a_1,...,a_k=v$，对于所有的 $0<i<k$，有 $u<v<a_i$。

如果 u 是另一个节点 v 中的所有 supreme vertex 中最小的，则 u 被称为 v 的 superior vertex。

询问每个节点有多少个点将其视为 superior vertex。

83.2 数据范围

多测，$1 \le T \le 10$

$1 \le N \le 100000$

$N-1 \le M \le 200000$

83.3 关键字 图论 半必经点

83.4 题解

裸半必经点……

83.5 复杂度

时间复杂度 $O(T(N+M))$

84、 Count on a treap

84.1 题意

这道题目要求你维护一个大根堆 Treap，要求支持：

1.插入一个关键字为 k，权值为 w 的点

2.删除一个关键字为 k 的点。

3.返回关键字分别为 u 和 v 两个节点的距离

保证任意时刻树中结点 key 和 weight 都是两两不同的。不会删除当前 Treap 中不存在的点。

84.2 数据范围

$1 \le N \le 200000$

84.3 关键字 数据结构

84.4 题解

此题询问两个节点的距离，不妨将其转化为 u 和 v 到根节点的距离的和减去两倍 LCA 到根节点的距离。

将节点按照权值从小到大排序。

u 和 v 的 LCA 就是它们之间关键字最大的点。

接下来的问题就是如何询问一个点 u 到根节点的距离。转化为考虑 u 节点祖先的数量，若一个点 v 是 u 的祖先，即 LCA(u,v)=v，所以 v 就是 u 到 v 之间关键字最大的点。直观来说就是从 u 开始向左右延伸的两个单调递增序列。

不妨考虑维护右边。

如果要将两个单调递增序列 A 和 B 合并。只需在 B 中二分到刚好比 A.Last 大的位置，将 B 的前面去掉，再将 A 和 B 接在一起即可。

我们对整个序列开一棵线段树，每个节点维护以第一个节点开始的递增序列。合并两个区间的递增序列按照上述方法合并。询问 u 只需从 u 开始往右一直并过去即可。

但时间上和空间上都不允许我们显式地维护每个节点的递增序列。我们也并不需要这么做，只需要记录两个序列的连接点即可。这样同样可以在线段树上对一个序列进行二分。

84.5 复杂度

时间复杂度 $O(n\log^2 n)$

85、 *Gangsters of Treeland*

85.1 题意

有 N 个城市，它们以树的方式连边，编号 0 到 N-1，城市 0 是首都。

初始时，每个城市都被一个帮会所控制。

村民在相邻的城市间移动，如果这两个城市不是隶属于同一个帮会，那么需要付出一个单位的代价。

每一年都有新的帮会涌入首都，他们会扩张自己的势力范围。

具体说来，他们会占据从首都到 u 路径上的所有城市（包括首都和 u）。

每次询问给定一个城市 u，定义 f(u)为以 u 为根的子树中所有节点到根节点的代价的平均值。

85.2 数据范围

多测，$1 \le T \le 15$

$1 \le N, Q \le 100000$，N 和 Q 的总和不超过 200000

85.3 关键字 数据结构 LCT

85.4 题解

每次新帮会的到来就相当于 LCT 一次 access 操作。

建立 LCT 模拟这个过程即可，只需在修改虚边时修改节点的代价。

85.5 复杂度

时间复杂度 $O(n\log n)$

86、 Queries on tree again

86.1 题意

有一个包含 N 个节点和 N 条边的无向连通图（不包含重边和自环）。

显然，这个图中仅包含一个环，且题目保证环的长度为奇数。

节点从 1 到 N 标号，每条边对应着一个整数权值。

定义节点 u 到 v 的最短路径为连接 u, v 的边数最少的路径。

你的任务是模拟以下两种操作：

1.对 u 到 v 的最短路径上的所有边的权值取相反数。

2.在 u 到 v 的最短路径上，找到一个连续的边的集合，使得集合中边的权值之和最大。

86.2 数据范围

$1 \le N, Q \le 100000$

86.3 关键字 数据结构 环套树 树链剖分

86.4 题解

环套树的经典题目，将环和树分开处理即可，对树进行树链剖分。

86.5 复杂度

时间复杂度 $O(n\log^2 n)$

87、 Course Selection

87.1 题意

课业计划共包含 N 项课程，每项课程都需要在 M 个学期里的某一个完成。

一些课程有前置课程：对于所有的 i(1<=i<=K),A[i]是 B[i]的前置课程。

相同的课程在不同的学期中可能会由不同的教授授课，不同的教授可能会影响在这一门课程上的表现。

我们给出数组 X 来描述这一信息。对于每项课程 i 和学期 j。X[i][j]表示在学期 j 选修课程 i 所能得到的期望分数。如果 X[i][j]=-1，则表示那个学期没有这个课程。

询问所能得到的分数的平均值的最大值。

87.2 数据范围

$1 \le N, M \le 100$

$0 \le K \le 100$

$-1 \le X[i][j] \le 100$

数据保证至少有一组解

87.3 关键字 网络流

87.4 题解

将第 i 项课程拆成 M 个点，我们要做到就是在 M 个点中选择一个点，并且一定要选一个点。

可以采用最小割，将这 M 个点首位相接成一条从 S 到 T 的路径，每条边的流量为-x[i][j]，此时保证了选择一个点，再将每条边的流量加上一个较大的数 maxx，此时保证了最多选一个。

至于前置课程(u,v)只需在点(u,i)和(v,i+1)之间连边即可。

87.5 复杂度

时间复杂度 $O(flow(NM, MK))$

88、Across the River

88.1 题意

目标是搭一座跨过河的桥。河是一条无限长的宽度为 W 的直线，所有在 xy-坐标系中符合 $0 \le y \le W$ 的点都属于这条河流。

河面上有 N 个木桩，还有 M 种可以用的木头圆盘，第 k 个木桩的坐标为 $P_k(X_k, Y_k)$。第 k 种圆盘半径为 R_k，每一块的价格为 C_k。

可以买任意多的圆盘，把它们放到河面上。每一个圆盘的中心都必须为某一个木桩的位置。注意，某些圆盘的一部分可以在地面上(y<0 或 W<y).

行人只能在直线 y=0 或直线 y=W 或圆盘上移动（相切的两者之间可以移动)。

询问搭一座桥的最小花费。

88.2 数据范围

$1 \le T \le 10$

$1 \le N, M \le 250$

88.3 关键字 最短路径 dp

88.4 题解

先将无用的圆盘去掉，即对于 i 存在 j 使得 $R_i < R_j, C_i > C_j$。

考虑这样的 dp：$f[i][j]$ 表示来到第 i 个木桩，上面放置了第 j 种圆盘。

$f[i][j]$ 能更新 $f[k][L]$，当且仅当 $R_j + R_L \ge dist(P_i, P_k)$。

用 SPFA 来实现上述的 dp，注意要优化连边。

88.5 复杂度

时间复杂度 $O(SPFA(NM, N^2M))$

89、Closest Points

89.1 题意

给出 N 个点 (x, y, z) 和 Q 个询问，每次询问离 (x_i, y_i, z_i) 最近的是哪个点。

89.2 数据范围

$N = Q = 50000$

89.3 关键字 KD-tree

89.4 题解

用 KD-tree 维护即可，可以限制深度卡时。

还有一点：对于一些特殊的数据，有许多次询问的答案都是同一个点，所以对于前 300 个使用 KD-tree 保证完全正确，后面的可以限制深度，再与前面的答案取最优。

90、Deleting numbers

90.1 题意

有一个数组 a，长度为 n。

每次操作可以选择两个数 v 和 t，满足 $v+t \le n+1$，且令 $k=\max\{k \mid v+kt \le n\}$，必须满足 $a[v]=a[v+t]=a[v+2t]=...=a[v+kt]$。然后删除这些数，新的数组有原来的 n-k 个数组成，相对位置不变。

求一种删除所有数的方案，步数越少越好。

90.2 数据范围

$1 \le n \le 100000$

90.3 关键字 贪心

90.4 题解

第一步贪心是将数量最多的数字拿出来，因为这些数字可以在最后一起删掉。

然后采取每次从最后的 100 个数里面取较优的操作。

91、To challenge or not

91.1 题意

从 M 个不同整数 B[i]中选出一部分，使得他们之间任意选三个数都不能组成等差数列。

91.2 数据范围

$10000 \le L \le 100000$，随机

$0.1 \le p \le 0.9$，随机

0 到 L 每个数都有 p 的概率加入 B

91.3 关键字 压位

91.4 题解

先将 B 数组随机打乱。

因为数据是随机的，所以直接从左到右对每个数进行判断，能加就加。

判断的时候用压位进行优化即可。

92、Stepping Average

92.1 题意

考虑以下的方法求出 N 个数的“迭代平均数”。

每次拿出其中任意两个数，并用他们的平均数取代他们，重复 N-1 次直到最后只剩一个数。我们叫这个剩下的数叫“迭代平均数”。值得注意的是，不同的合并顺序最后会导出不同的“迭代平均数”。

你要做的是：给你 N 个数。找到一个合并的顺序使得“迭代平均数”尽量接近给定的数 K。

92.2 数据范围

多测，$T = 10$

$N = 1000$

92.3 关键字 贪心

92.4 题解

第一个想法就是每次将离 K 最远的两个数取平均，然而这样对于随机数据效果并不好，主要是比 K 大和比 K 小的数放在一起考虑会出现一些小问题。

同样试过将其分开考虑，然而由于估价函数弄得十分不好比暴力还差……

换个思路，考虑最小的数和最大的数 A 和 B，设 $mid = \frac{A+B}{2}$，如果 mid 与 K 比较接近，则立刻合并数 A 和 B。否则如果 $mid < K$，则将 A 与 B 放在最后合并，将 A 除去，新的 K 变为 $2K - A$； $mid > K$ 情况类似。

事实证明这样做对于随机数据效果十分的好。

93、Fault Tolerance

93.1 题意

有 N 个元素和 M 个包含这些元素的集合。保证一开始每个元素都能通过一些集合异或得到。问最少删掉多少个集合，使得存在一个元素不可能通过一些集合异或得到。

93.2 数据范围

$50 \le N \le 200$

$2N \le M \le 2000$

93.3 关键字 线性代数 贪心 随机化

93.4 题解

我们将集合一行一行排下来，如果第 i 列（就是每个集合中 i 元素的 01 状态）可

以被其他一些列 $a_1,a_2\ldots$ 异或得到，那么元素 i 肯定不能通过一些集合异或得到。因为如果异或出来第 i 列为 1，则列 $a_1,a_2\ldots$ 异或起来为 1，意味着里面肯定有一个 1。

我们按照这样的方法构造一组方案。

我们有一个初步的思路：将列进行各种异或操作，每次将操作的列的 1 所在的行全部删掉，该列就会被其他列异或得到。

首先贪心，对于每一列 i，找到另一列 j，使得 $a_i \wedge a_j$ 的 1 最少，$a_i \wedge= a_j$

然后随机化，随机一列 i，再随机该列上的一个 1，设为第 j 位，然后对于所有的列 k 满足第 j 位为 1 的都令 $a_j \wedge= a_i$，更新一下答案。

94、 *Maximum Sub-rectangle in Matrix*

94.1 题意

给出一个 H*W 的整数矩阵 A，行标号 0 到 H-1，列标号 0 到 W-1，求一个子矩阵使其中元素之和尽量大，注意这个子矩阵不要求连续。

94.2 数据范围

$200 \le H,W \le 300$

94.3 关键字 贪心 随机化

94.4 题解

如果选择的行确定了，那么列肯定也确定了，即为和大于 0 的列。

考虑用以上的贪心来不断确定行和列，即由行确定列，再由列推到行，以此类推。

再由上面的算法加随机化，每次选择大于 0 的行（列）后，随机加或减几行。

95、 *Chef and Painting*

95.1 题意

游戏的棋盘是一个 N 行 M 列的矩阵，行从 1 到 N 标号，列从 1 到 M 标号。

初始时，棋盘上有一些黑色的格子，其它的格子都是白色的。

每一步，可以选择任意一个白色的格子和一个方向（左-右或上-下）

如果选择了左-右，那么从选定的白色格子开始，向左边依次把格子涂成红色，直到到达了棋盘边界或者到达了某个之前已经被染成了黑色或红色的格子时停止涂色；然后再从选定的格子开始，向右边进行同样的涂色操作。

如果选择了上-下，也与左-右类似，在选定的格子的上下方向上颜色。

游戏的目标是用尽可能少的步数把所有原来的白色的格子都涂成红色。

95.2 数据范围

$1 \le N, M \le 100$

$1 \le K \le \min(NM-1, 3000)$

数据随机

95.3 关键字 随机化

95.4 题解

有一个初步的想法：从上到下，从左到右，每个格子选择左-右还是上-下较优的进行操作。这个方法实际上效果也不错了。

可以再进行随机化优化：每次八个方向随机一下（从上到下或从下到上，从左到右或从右到左，先上下再左右或先左右再上下）然后对于那个方向的第一个白色格子进行判断即可。

再和暴力取个最优值。

96、*Killing Gs*

96.1 题意

这里有 n 只 G，打算用杀虫剂杀死所有的 G。现在有 m 种可用的杀虫剂，第 j 种杀虫剂的价格是 Cj。但是通过统计我们知道，如果使用第 j 种杀虫剂，第 i 只 G 将会有 Pi,j%的概率死亡（这些概率是独立的）。

我们会一种接一种地使用杀虫剂（每种只能使用一次），选择一种花钱尽量少的方案，使得最后所有的 G 都有至少 90%的概率死亡。

96.2 数据范围

多测 $T = 10$

$50 \le n, m \le 200$

96.3 关键字 贪心 随机化 模拟退火

96.4 题解

考虑将 p[i][j]对 10 取对数，记为 w[i][j]。问题转化选取一些杀虫剂 $a_1, a_2 \ldots a_k$，

使得对于所有的$1 \le i \le N$，都有$\sum_{j=1}^{k} w[a_i][j] \ge 1$。

于是贪心地每次取贡献最大的杀虫剂，得到一个初步方案。

再经过模拟退火微调即可。

97、Similar Graphs

97.1 题意

给出两个 N 个点的无向图，它们的连边方式分别用 A 和 B 表示。我们希望对这两幅图的节点进行重标号，使得尽量多的$A[i][j] = B[i][j]$。

97.2 数据范围

多测，$T = 5$

$30 \le N \le 75$

97.3 关键字 随机化 贪心 KM 算法

97.4 题解

可以注意到只对第二幅图进行重编号即可。

每次随机 5 到 10 个点，用 KM 算法对其进行重编号，多次重复上述过程即可。

98、Kali and Devtas

98.1 题意

给定二维欧氏平面内的 N 个点，你需要返回这些点的一个生成树，使得 Ci 的最大值最小。

Ci 的定义是：对于每个点，设在生成树中与其相邻的点中最远的点的距离为 R，那么以该点为圆心，半径 R 以内的的点的 Ci 全部增加 1（包括自身）。

98.2 数据范围

$1 \le T \le 100$

$3 \le N \le 400$

98.3 关键字 最小生成树

98.4 题解

弄个最小生成树即可……