

2026 年清华大学学生程序设计竞赛 暨高校邀请赛

THUPC 2026

决赛

时间：2026 年 4 月 12 日 10:00 ~ 15:00

目 录

A. 庆典门票 / Ticket	2
B. 展览区寻宝 / Hunt	4
C. 盲盒抽奖 / Unboxing	6
D. 幻光留影 / Photo	8
E. 积木消除游戏 / Cubes	12
F. 棋盘对弈游戏 / Tactics	14
G. 论文审稿 / Essay	18
H. 社交网络 / Network	20
I. 年鉴整理 / Yearbook	22
J. 供电网络 / Electricity	24
K. 流光解密 / Aura	28
L. 星图重绘 / Redrawing	34
M. 庆典留念 / Souvenir	38

A. 庆典门票 / Ticket

【题目背景】

为了庆祝 THUPC 十周年，小 T 和小 S 正在筹备一场盛大的十周年庆典！

庆典的首要准备工作是确定主会场。他们选定了一个房间作为主会场，并用森林墙纸、硕大无朋的大蒜玩偶、双“囍”壁挂（今年恰好也是清华 $(55)_{22}$ 周年校庆）和羽毛摆件把它装饰得整整齐齐。然而，兢兢业业的小 T 发现了一个棘手的问题：赫然挂在主会场前的门牌上的房间号看起来不够整齐。小 S 弱弱地提出，可以通过转换进制的方式，让房间号变得整齐。在尝试的过程中，他们发现能让房间号达到整齐效果的进制转换方案远不止一种。于是，小 T 和小 S 决定将这个有趣的房间号设计过程作为庆典的入场挑战，交由前来参会的大家解答。

【题目描述】

小 T 和小 S 确定的主会场房间号在十进制下为 n 。小 T 对整齐的房间号表示方式给出了如下定义：对于正整数 $b, p \geq 2$ ，若房间号 n 的 b 进制表示恰好由若干段长度为 p 的相同数字拼接而成，则认为 (b, p) 是一种整齐的表达方式。

形式化地，设 n 的 b 进制表示为 $\overline{d_{k-1}d_{k-2}\dots d_1d_0}$ ，若存在正整数 c 使得总位数 $k = cp$ ，并且对于所有的 $0 \leq i < c$ ，均满足 $d_{ip} = d_{ip+1} = \dots = d_{(i+1)p-1}$ ，则 (b, p) 是一种整齐的表达方式。

例如，若房间号为 2233 或 3355，则 $(10, 2)$ 是一种整齐的表达方式；若房间号为 1111，则 $(10, 2), (10, 4)$ 是两种不同的整齐的表达方式；若房间号为 6737151（十六进制表示为 66CCFF），则 $(16, 2)$ 是一种整齐的表达方式。

为了顺利赢取入场门票，你需要回答小 T 和小 S 的问题：主会场的房间号一共存在多少种整齐的表达方式？

【输入格式】

每个测试点中包含多组测试数据。输入的第一行包含一个正整数 T ($1 \leq T \leq 10^3$)，表示数据组数。对于每组测试数据：

- 第一行包含一个正整数 n ($1 \leq n \leq 10^{12}$)，表示主会场的房间号。
保证所有测试数据中 n 的和不超过 10^{12} 。

【输出格式】

对于每组测试数据，输出一行一个非负整数，表示答案。

【样例 1 输入】

```
1 10
2 1
3 2
4 115
5 1111
6 2233
7 3355
8 191970
9 6737151
10 102934760424
11 618111100000
```

【样例 1 输出】

```
1 0
2 0
3 2
4 4
5 5
6 5
7 24
8 9
9 17
10 144
```

【样例 1 解释】

对于第三组测试数据， $115 = (55)_{22} = (11)_{114}$ ，因此所有整齐的表达方式分别为 $(22, 2), (114, 2)$ 。

对于第四组测试数据，所有整齐的表达方式分别为 $(10, 2), (10, 4), (100, 2), (1110, 2)$ 。

B. 展览区寻宝 / Hunt

【题目背景】

在确定好主会场的位置后，小 T 和小 S 开始着手布置庆典会场。他们在通往主会场的必经之路上，设置了一块宽阔的展览区，用于展现十年 THUPC 的精彩画面。

小 T 将展览区规划成了一个巨大的网格，最外围以及内部的部分格子被设为展览墙。为了方便大家沿动线参观，他将所有的展览墙精心设计为了四连通的结构。

为了让参观过程更具趣味性，小 S 决定在这里举办一场寻宝活动。

【题目描述】

小 T 将展览区规划成一个大小为 $n \times n$ 的二维网格。网格的最外围由一圈展览墙包围，即横坐标或纵坐标等于 0 或 $n + 1$ 的所有格子均为展览墙格子。此外，展览区内部还散布着 m 个展览墙格子，其中第 i ($1 \leq i \leq m$) 个的坐标为 (x_i, y_i) 。保证所有的展览墙格子之间是四连通的。

经过实地布景测试，小 T 总结出了在网格间移动的耗时规律。具体而言，在格子间有以下两种移动方式：

- 沿上下左右方向移动一格，即从 (x, y) 移动到 $(x-1, y), (x+1, y), (x, y-1), (x, y+1)$ 中的某一个相邻格子，需要消耗 2 个单位时间。
- 沿对角线方向移动一格，即从 (x, y) 移动到 $(x-1, y-1), (x-1, y+1), (x+1, y-1), (x+1, y+1)$ 中的某一个对角格子，需要消耗 3 个单位时间。

当然，移动的目标位置不能是展览墙格子。注意：沿对角线方向移动时，可以直接从两个对角展览墙格子之间的缝隙穿过。例如，即使 $(x, y+1)$ 与 $(x+1, y)$ 均为展览墙格子，依然可以消耗 3 个单位时间，直接从 (x, y) 沿对角线移动到 $(x+1, y+1)$ 。

小 S 在展览区中总计布置了 q 个宝藏。对于第 i ($1 \leq i \leq q$) 个宝藏，她会公布它的位置 (tx_i, ty_i) ，而公布时你所处的位置为 (sx_i, sy_i) 。为了以最快速度夺得各个宝藏，你需要计算出，从你所处的位置移动到宝藏所在位置的最短耗时。

【输入格式】

输入的第一行包含三个正整数 n, m, q ($1 \leq n \leq 10^5, 1 \leq m, q \leq 3 \times 10^5$)。

接下来 m 行，第 i ($1 \leq i \leq m$) 行包含两个正整数 x_i, y_i ($1 \leq x_i, y_i \leq n$)，表示第 i 个展览墙格子的坐标。保证所有 (x_i, y_i) 两两不同。

接下来 q 行，第 i ($1 \leq i \leq q$) 行包含四个正整数 sx_i, sy_i, tx_i, ty_i ($1 \leq sx_i, sy_i, tx_i, ty_i \leq n$)，表示公布第 i 个宝藏时你所处的位置与宝藏所在的位置。保证 $(sx_i, sy_i), (tx_i, ty_i)$ 均不为展览墙格子。

【输出格式】

输出 q 行，每行一个整数表示答案。特别地，若你无法移动到宝藏所在的位置，则输出 -1 。

【样例 1 输入】

```
1 4 4 5
2 2 1
3 2 2
4 3 2
5 3 3
6 1 1 1 2
7 1 1 3 1
8 4 1 1 4
9 4 4 1 1
10 2 3 3 1
```

【样例 1 输出】

```
1 2
2 16
3 11
4 10
5 11
```

【样例 1 解释】

对于第二个宝藏，你可以沿着以下路径移动： $(1,1) \rightarrow (1,2) \rightarrow (2,3) \rightarrow (3,4) \rightarrow (4,3) \rightarrow (4,2) \rightarrow (3,1)$ ，总耗时为 $2 + 3 + 3 + 3 + 2 + 3 = 16$ 。

C. 盲盒抽奖 / Unboxing

【题目背景】

通过了门票考验并从展览区入场后，大家便来到了热闹的互动大厅。在这里，小 T 和小 S 特意设置了盲盒抽奖环节。

抽奖摊位上的盲盒会随着大家的到来而陆续上架。小 T 为此制定了一套别出心裁的兑奖规则：每个人都可以从台上挑出一部分盲盒，并按这些盲盒原有的先后顺序两两配对。只有当这组盲盒背后的隐藏数字满足特定的运算条件时，配对才算成功并能兑换相应的奖品。

【题目描述】

活动现场总计会陆续上架 n 个盲盒，它们背后的隐藏数字分别为 $a_1, a_2, \dots, a_n$ 。

在抽奖环节中，若当前台上已经展示了前 k 个盲盒，则参与者可以从中挑选出偶数个盲盒（设其对应原序列的下标为 $1 \leq i_1 < i_2 < \dots < i_{2t} \leq k$ ），并将它们按顺序依次配对为 t 组，即 $(a_{i_1}, a_{i_2}), (a_{i_3}, a_{i_4}), \dots, (a_{i_{2t-1}}, a_{i_{2t}})$ 。对于任意一对被挑选出的盲盒，设其背后的隐藏数字分别为 x 和 y ，则兑奖的条件是： x 与 y 的二进制按位异或结果必须严格小于小 T 提前设定的幸运阈值 m 。满足该条件的每一对盲盒均可算作有效配对，并兑换一个奖品。

作为热情的参与者，你也体验了这场抽奖，但十分遗憾，你选出的盲盒无一满足兑奖条件。为了安慰运气不佳的你，小 S 向你抛出了一个挑战：如果你能正确回答她提出的问题，她便会直接送你一份特别的十周年大奖。

小 S 的问题是：对于每一个 $k \in [1, n]$ ，当台上恰好展示了前 k 个盲盒时，能够兑换的最大奖项总数，是否严格大于仅展示前 $k - 1$ 个盲盒时的最大数量？

【输入格式】

输入的第一行包含两个正整数 n, m ($1 \leq n \leq 5 \times 10^6, 2 \leq m \leq 10^8$)，分别表示盲盒的总数与小 T 提前设定的幸运阈值。

输入的第二行包含 n 个正整数 $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^8$)，分别表示每个盲盒背后的隐藏数字。

【输出格式】

输出一行一个长度为 n 的字符串。对于每一个 $k \in [1, n]$ ，若展示前 k 个盲盒能兑换的最大奖项数严格大于仅展示前 $k - 1$ 个盲盒时的最大奖项数，则字符串的第 k 位为 Y，否则为 N。

【样例 1 输入】

```
1 5 4
2 1 2 5 4 3
```

【样例 1 输出】

```
1 NYNYN
```

【提示】

本题输入量较大，建议使用较为快速的输入方式。

D. 幻光留影 / Photo

【题目背景】

为了给大家留下独特的十周年纪念，小 T 和小 S 在主会场中布置了一个大型的交互艺术装置“幻光留影”。

这个装置由许多悬浮的影像结点组成，通过光束连接成了一棵树的形态，并为每个结点配置了不同颜色的光影滤镜。你来到了控制台前，发现这里可以自由互动：每当调节控制台上的参数，系统会临时只激活特定编号区间内的光束，这会让原本完整的留影网络分裂成若干个独立的连通子区域。

充满巧思的是，同一连通子区域中带有同种滤镜的结点之间会发生光学干涉：当某种颜色的滤镜在该连通子区域内出现了偶数次时，它们的光芒将相互抵消变得不可见；而当其出现次数为奇数时，则会汇聚成清晰可见的色彩。

光影的破碎与重组十分迷人，你对其中隐藏的结构变化产生了浓厚的兴趣：在每一次这样的局部激活下，所有分裂出来的连通区域中，各自包含了多少种不同的可见滤镜呢？

【题目描述】

幻光留影装置可抽象为一棵包含 n 个结点的树，每个结点代表一个影像结点，结点 i ($1 \leq i \leq n$) 的滤镜颜色为 c_i 。连接结点的光束共有 $n-1$ 条，并依次编号为 1 到 $n-1$ 。

在你尝试探索规律的过程中，记录了 m 次动态效果测试的数据。对于每一次测试，你指定了一个光束的编号区间 $[l, r]$ 。假设此时只激活编号落在该区间内的光束（断开编号不在该区间内的其他光束），原先连通的整个网络便会分裂成若干个独立的连通块。

为了更深入地分析其中的变化，你需要针对每一次测试计算出：所有连通块中，各自包含的可见滤镜（即出现次数为奇数的滤镜颜色）种类数的总和。

【输入格式】

输入的第一行包含两个正整数 n, m ($1 \leq n \leq 3 \times 10^5, 1 \leq m \leq 10^6$)，分别表示影像结点的数量与测试次数。

输入的第二行包含 n 个正整数 $c_1, c_2, \dots, c_n$ ($1 \leq c_i \leq n$)，分别表示每个影像结点的滤镜颜色。

接下来 $n-1$ 行，第 i ($1 \leq i \leq n-1$) 行包含两个正整数 u_i, v_i ($1 \leq u_i, v_i \leq n$)，表示编号为 i 的光束连接的两个结点。

接下来 m 行，每行包含两个正整数 l, r ($1 \leq l \leq r \leq n-1$)，表示一次测试的光束编号区间。

【输出格式】

输出 m 行，每行一个非负整数，依次表示每次测试中所有独立连通块各自包含的可见滤镜种类数的总和。

【样例 1 输入】

```
1 12 13
2 2 4 1 2 3 4 2 4 3 3 3 6
3 3 5
4 9 1
5 5 11
6 4 9
7 1 12
8 2 1
9 10 8
10 11 10
11 8 4
12 6 11
13 7 6
14 1 3
15 2 2
16 6 6
17 9 11
18 6 11
19 1 4
20 8 10
21 1 11
22 5 10
23 4 7
24 1 10
25 6 8
26 11 11
```

【样例 1 输出】

```
1 10
2 12
```

3	12
4	12
5	6
6	8
7	10
8	4
9	8
10	12
11	4
12	10
13	12

此页刻意留白

E. 积木消除游戏 / Cubes

【题目背景】

欣赏完绚丽的幻光留影，大家又被不远处的积木消除小游戏区吸引了目光。

桌面上整齐排列着五颜六色的积木。小 T 和小 S 作为摊主，各自提供了一个能够批量消除积木的魔法筛网。游戏规则很简单：大家可以反复使用这两个筛网进行消除，最终根据桌面上剩余总积木数量进行排名。

【题目描述】

桌面上整齐排列着 n 堆积木，第 i ($1 \leq i \leq n$) 堆的初始数量为 a_i 。

小 T 和小 S 分别提供了网眼大小为 p, q 的两个魔法筛网，能将覆盖的积木堆按对应的模数取余，从而将积木批量消除。在自然展开时，这两个筛网都恰好跨越 k 堆积木的宽度。它们具有特殊的弹性，可以向两端自由拉伸以覆盖更长的范围，但无法向内压缩收拢。魔法筛网的使用方式如下：

- 选定一段长度至少为 k 的连续的积木区间 $[l, r]$ 并铺上筛网；
- 从两个魔法筛网中任选一个，即选定 $m \in \{p, q\}$ ；
- 对于区间 $[l, r]$ 内的每一堆积木，将其数量对 m 取余，即令 $a_i \leftarrow a_i \bmod m$ 。

既然参与了这场游戏，你自然不满足于平庸的成绩。为了在排行榜上拔得头筹，你想知道，通过反复使用任意次数的魔法筛网，最终桌面上剩余的积木总数（即 $\sum_{i=1}^n a_i$ ）最少能被消除到多少？

【输入格式】

每个测试点中包含多组测试数据。输入的第一行包含一个正整数 T ($1 \leq T \leq 10^3$)，表示数据组数。对于每组测试数据：

- 第一行包含四个正整数 n, k, p, q ($1 \leq k \leq n \leq 10^5$, $1 \leq p < q \leq 10^9$)，分别表示积木的堆数、筛网自然展开时跨越的积木堆数，以及两个魔法筛网的网眼大小。
- 第二行包含 n 个正整数 $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$)，分别表示每堆积木的初始数量。

保证所有测试数据中 n 的和不超过 10^5 。

【输出格式】

对于每组测试数据，输出一行一个非负整数，表示桌面上剩余总积木数量的最小值。

【样例 1 输入】

```
1 6
2 1 1 3 4
3 2026
4 3 2 10 20
5 31 41 59
6 4 3 3 4
7 1 2 3 4
8 6 4 9 20
9 18 27 180 9 45 99
10 7 4 3 5
11 6 7 14 12 100 78 4
12 9 4 244 353
13 9982 4435 3998 2443 5399 8244 3539 9824 4353
```

【样例 1 输出】

```
1 1
2 11
3 3
4 0
5 4
6 569
```

【样例 1 解释】

对于第二组测试数据，一种能使桌面上剩余积木总数达到最小值 11 的操作方式如下：

- 选定区间 $[1, 4]$ 并使用网眼大小为 10 的魔法筛网，剩余的积木数量变为 $[1, 1, 9]$ 。

对于第三组测试数据，一种能使桌面上剩余积木总数达到最小值 3 的操作方式如下：

- 选定区间 $[2, 4]$ 并使用网眼大小为 4 的魔法筛网，剩余的积木数量变为 $[1, 2, 3, 0]$ ；
- 选定区间 $[1, 3]$ 并使用网眼大小为 3 的魔法筛网，剩余的积木数量变为 $[1, 2, 0, 0]$ 。

F. 棋盘对弈游戏 / Tactics

【题目背景】

在单人操作的积木消除小游戏后，小 T 和小 S 还准备了一项双人互动的棋盘对弈小游戏。

游戏在一套特制的十周年纪念棋盘上进行。棋盘上面分布着带有分值的格子，参与的双方需要轮流控制棋子向前跳跃，抢占格子上的分数，以最大化自己和对手分数的差值。

【题目描述】

这套特制的棋盘由 n 个格子组成，其中格子 i ($3 \leq i \leq n$) 的分值为正整数 a_i 。

你决定与你的队友进行一场对弈。游戏开始时，你占据了格子 1 并放上了自己的棋子，而你的队友占据了格子 2 并放上了他的棋子，且此时仅有这两个格子被占据。双方的初始得分均为零。

游戏由你率先行动，之后双方轮流进行。在每次行动中，设当前行动玩家的棋子位于格子 x ，则该玩家必须选择一个步数 $d \in \{1, 2, 3, 4\}$ ，满足 $x + d \leq n$ ，且格子 $x + d$ 尚未被占据，然后将自己的得分加上该格子对应的分值 a_{x+d} ，并将其棋子向前跳跃至格子 $x + d$ 。跳跃完成后，该玩家将永久占据这一新格子。特别地，若不存在任何合法的跳跃步数，该玩家本回合将无法行动并直接跳过。当双方均无法行动时，游戏结束。

显然，擅长博弈的你和你的队友都足够聪明，均会在对弈中采取最优策略。为了提前推演对局结果，你需要计算出在游戏结束时，你的总得分减去你的队友的总得分的值。

【输入格式】

每个测试点中包含多组测试数据。输入的第一行包含一个正整数 T ($1 \leq T \leq 10^3$)，表示数据组数。对于每组测试数据：

- 第一行包含一个正整数 n ($6 \leq n \leq 10^5$)，表示棋盘的格子数量。
- 第二行包含 $n - 2$ 个正整数 $a_3, a_4, \dots, a_n$ ($1 \leq a_k \leq 10^9$)，分别表示每个格子的分值。

保证所有测试数据中 n 的和不超过 10^5 。

【输出格式】

对于每组测试数据，输出一行一个整数，表示你的总得分减去你的队友的总得分的值。

【样例 1 输入】

```

1 6
2 6
3 1 6 3 4
4 10
5 1 1 1 1 1 1 1 1
6 10
7 1 1 1 1 1 1 1 10
8 9
9 1 1 1 1 1 1 10
10 8
11 10 1 1 1 1 100
12 10
13 1000000000 1 1000000000 1 1000000000 1 1000000000 1

```

【样例 1 输出】

```

1 5
2 0
3 -7
4 8
5 90
6 1000000000

```

【样例 1 解释】

以下用一个长度为 n 的字符串表示对局结果，其中字符 $.$ 表示该格子未被任何人占据，字符 O 表示该格子被你占据，字符 X 表示该格子被你的队友占据。

对于第一组测试数据，在第一次行动中，你有以下三种选择：

- 选择步数 $d = 2$ ，将棋子跳跃至格子 3，则对局结果为 $OXOXXO$ ，得分差为 -6 ；
- 选择步数 $d = 3$ ，将棋子跳跃至格子 4，则对局结果为 $OX.OOX$ ，得分差为 5 ；
- 选择步数 $d = 4$ ，将棋子跳跃至格子 5，则对局结果为 $OX..OX$ ，得分差为 -1 。

因此对局结果为 $OX.OOX$ ，得分差为 5 。

对于第二组测试数据，一种可能的对局结果为 $OXOXOXOX$ ，得分差为 0 。

对于第三组测试数据，一种可能的对局结果为 $OX..OX000X$ ，得分差为 -7 。

对于第四组测试数据，一种可能的对局结果为 $OX..OXXXO$ ，得分差为 8 。

对于第五组测试数据，一种可能的对局结果为 OXXXOX00，得分差为 90。

对于第六组测试数据，一种可能的对局结果为 OX..OX0.X0，得分差为 1000000000。

【样例 2 输入】

```
1 6
2 9
3 7 6 2 2 5 8 7
4 10
5 8 26 18 1 11 9 15 9
6 11
7 8 3 9 2 3 4 8 8 7
8 12
9 5 6 5 3 1 2 1 1 5 4
10 15
11 6 6 7 2 2 2 5 2 2 4 7 7 7
12 18
13 7 4 5 1 2 6 7 5 7 3 7 3 6 5 6 6
```

【样例 2 输出】

```
1 5
2 13
3 8
4 3
5 9
6 9
```

此页刻意留白

G. 论文审稿 / Essay

【题目背景】

诸多趣味项目告一段落，会场迎来了中场休息的茶话会。众人闲谈间聊起了科研的艰辛与论文投稿的不易：小 T 感慨自己每次投稿后都会不断撤回重投、反复打磨论文；而作为审稿人的小 S 则表示，自己审阅时必然会严格把关，尽可能筛除手上的稿件。

根据这番交流，小 T 和小 S 突发奇想：如果遇到一位极其苛刻的审稿人，论文的发表之路会演变成什么样？于是，他们顺势提出了一个博弈模型。在这个模型中，双方将分别扮演投稿人与审稿人，围绕一个漫长的审稿队列展开博弈。队列中混杂着大量他人的稿件，其中仅有一篇是投稿人投出的。投稿人希望让自己的这篇论文在队列中反复循环打磨，以尽可能提升其最终发表时的学术分值；而审稿人则会严苛地筛查，寻找机会直接毙掉这篇论文，让投稿人颗粒无收。

【题目描述】

审稿队列由 n 篇论文组成，其中第 m 篇是投稿人投出的论文。初始时，该篇论文的学术分值为 1。

博弈由**投稿人**率先行动，之后双方轮流进行。每次行动的一方需要依次执行以下操作：

1. 从审稿队列首部依次取出 k 篇论文。若队列中剩余论文不足 k 篇，则将它们全部取出；
2. 在这批取出的论文中，选择将其中的 0 或 1 篇论文发表或者永久拒稿，即将其彻底移除；
3. 将**所有**未被移除的论文以**任意**顺序重新插入审稿队列末尾。由于审稿队列全程公开，双方均能准确知晓重新插入后每篇论文的新位置。

投稿人的最终得分将由他投出的论文的结果决定：

- 每次行动中，若该篇论文未被移除且被重新插入审稿队列末尾，其学术分值将会增加 1；
- 若**投稿人**在某次操作中主动移除了该篇论文，则该篇论文成功发表，博弈立即结束，投稿人获得此时的学术分值；
- 若**审稿人**在某次操作中移除了该篇论文，则该篇论文被永久拒稿，博弈立即结束，投稿人获得 0 分。

投稿人的目标自然是最大化得分，而审稿人的目标是最小化投稿人的得分。

小 T 和小 S 目睹了你在前一场对弈游戏中的优异表现，于是邀请你来分析这个有趣的博弈模型。你需要计算出，当投稿人与审稿人均采取最优策略时，投稿人的最终得分。要是你能帮他们算出结果，说不定他们会乐意亲自帮你修改论文哦！

【输入格式】

每个测试点中包含多组测试数据。输入的第一行包含一个正整数 T ($1 \leq T \leq 50$), 表示数据组数。对于每组测试数据:

- 第一行包含三个正整数 n, k, m ($1 \leq n, k \leq 10^9, 1 \leq m \leq n$), 分别表示审稿队列的长度、每次取出的论文数量, 与初始时投稿人投出的论文所处的位置。

【输出格式】

对于每组测试数据, 输出一行一个整数, 表示投稿人的最终得分。特别地, 如果博弈不会结束, 则输出 -1 。

【样例 1 输入】

```
1 6
2 3 2 1
3 5 3 4
4 10 3 1
5 7 3 7
6 817247666 7237 327476688
7 610723117 332458760 292738094
```

【样例 1 输出】

```
1 2
2 0
3 2
4 4
5 3470
6 278264358
```

H. 社交网络 / Network

【题目背景】

讨论完令人头疼的科研与审稿，茶话会的话题逐渐转向了日常的社交。小 T 兴致勃勃地分享了他的一个有趣发现：在某一常用的社交平台上，用户彼此间的关注关系恰好都是严格的单向关注，即不存在两名用户互相关注的情况。

这个奇妙的现象立刻引起了大家的讨论。小 S 顺势提取了该网络的部分统计数据，收集到了每位用户的关注数与粉丝数。遗憾的是，她在传阅数据的过程中不小心弄丢了一部分，最后只留下了一个由若干正整数构成的集合。小 S 发现，对于集合中的每一个元素，总能找到至少一个用户，其关注数或粉丝数恰好等于该元素。

由于平台完整的关注结构对外保密，具体的社交关系网已无从知晓。为了验证这些残存数据的合理性，大家纷纷拿起了纸笔，尝试着还原出符合条件的网络。为了增添几分趣味与竞技性，大家甚至就此展开了一场小小的比赛，看谁能构造出总关注数最少的社交网络。

【题目描述】

在这一社交平台上，共有 n 位用户。小 S 收集到了一个大小为 m 的数字集合 $\{c_1, \dots, c_m\}$ 。根据这些信息，一个可能的关注网络可以抽象为一张满足以下条件的有向图 $G = (V, E)$ ：

- 包含 n 位用户，即顶点集合 $V = \{1, 2, \dots, n\}$ ；
- 不存在自己关注自己或重复关注的情况，即图 G 不含自环与重边；
- 所有关注关系均为严格的单向关注，即对于任意有向边 $(u, v) \in E$ ，均满足 $(v, u) \notin E$ ；
- 对于集合中的每一个元素 c_i ($1 \leq i \leq m$)，图 G 中都至少存在一个顶点，其出度（对应关注数）或入度（对应粉丝数）恰好等于 c_i 。

你需要根据小 S 收集到的信息，还原出一个总关注数最少（即图 G 边数最小）的关注网络。

【输入格式】

输入的第一行包含一个非负整数 $o \in \{0, 1\}$ ，表示输出的参数。

输入的第二行包含两个正整数 n, m ($1 \leq m < n \leq 10^6$)，表示用户数量与小 S 收集到的集合的大小。保证若 $o = 0$ ，则 $n \leq 2 \times 10^3$ 。

输入的第三行包含 m 个两两不同的正整数 $c_1, c_2, \dots, c_m$ ($1 \leq c_i \leq n - 1$)，表示小 S 收集到的集合中的元素。

【输出格式】

输出一行一个正整数 k ，表示所有可能的关注网络中总关注数的最小值。

若 $o = 0$ ，则接下来输出 k 行，每行包含两个正整数 u, v ($1 \leq u, v \leq n$)，表示用户 u 关注了用户 v ，即 $(u, v) \in E$ 。

【样例 1 输入】

```
1 0
2 5 4
3 3 1 4 2
```

【样例 1 输出】

```
1 7
2 3 2
3 4 1
4 3 4
5 4 5
6 3 5
7 4 2
8 3 1
```

I. 年鉴整理 / Yearbook

【题目背景】

茶话会临近尾声，作为庆典十周年专属的怀旧环节，小 T 特意找出了历届 THUPC 的珍贵档案——一排记录了十年点滴的赛事年鉴，供大家传阅。

翻阅过后，两人准备将年鉴放回书架。由于岁月流逝，年鉴的破损程度参差不齐。心思细腻的小 S 提议，在放回时将它们按破损程度严格递增重新排序，以此直观地展现时间沉淀的痕迹。然而，年鉴的纸张已经十分脆弱，每次只能极其小心地将其中的一本向前平移一位，且这无可避免地会略微增加该年鉴的破损度。更棘手的是，留给他们的整理时间十分有限。小 S 想知道，能否在有限的操作次数内，将这些年鉴放回书架并整理得井然有序呢？

【题目描述】

书架上摆放着 n 本年鉴。初始时，第 i ($1 \leq i \leq n$) 本年鉴的破损度为 a_i 。

每次移动时，首先需要选择一个位置 p ($1 \leq p \leq n-1$)，然后将第 $p+1$ 本年鉴向前移动至第 p 本年鉴前，移动后它的破损度将会增加 1。

由于时间有限，总共只能进行不超过 $n^2 - n$ 次移动。作为众多翻阅年鉴的参与者之一，你需要帮助小 S 规划出一组具体的移动方案，使得最终书架上的年鉴破损度从左到右严格递增。

【输入格式】

每个测试点中包含多组测试数据。输入的第一行包含一个正整数 T ($1 \leq T \leq 10$)，表示数据组数。对于每组测试数据：

- 第一行包含一个正整数 n ($2 \leq n \leq 500$)，表示年鉴的数量。
- 第二行包含 n 个正整数 $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$)，分别表示初始时每一本年鉴的破损度。

【输出格式】

对于每组测试数据，若存在可行的移动方案：

- 第一行输出一个非负整数 k ($0 \leq k \leq n^2 - n$)，表示移动次数。
- 第二行输出 k 个正整数 $p_1, p_2, \dots, p_k$ ($1 \leq p_i \leq n-1$)，分别表示每次移动选定的位置。

若无法使得最终书架上的年鉴破损度从左到右严格递增，则仅输出一行一个整数 -1 。

【样例 1 输入】

```
1 3
2 2
3 1 2
4 2
5 2 1
6 3
7 4 5 1
```

【样例 1 输出】

```
1 0
2
3 -1
4 2
5 2 1
```

【样例 1 解释】

对于第一组测试数据，初始时书架上的年鉴破损度从左到右就是严格递增的。

对于第二组测试数据，可以证明，无法在 $n^2 - n$ 次移动内使得最终书架上的年鉴破损度从左到右严格递增。

对于第三组测试数据，一种可行的移动方案如下：

- 第一次移动选择位置 2，所有年鉴的破损度变为 $[4, 2, 5]$ ；
- 第二次移动选择位置 1，所有年鉴的破损度变为 $[3, 4, 5]$ 。

J. 供电网络 / Electricity

【题目背景】

整理完年鉴后，会场准备开启夜间活动的灯光特效。然而，由于后台的供电网络尚未配置完成，备受期待的全息投影仪却迟迟未能亮起。

整个电网由众多中继节点组成，这些节点必须分别接入两个并行的主供电模块。根据电气安全协议，处于同一模块内的相邻节点极易产生相位共振。因此，系统对每个节点在所属模块内的邻点总数，有着严格的奇偶性限制。

面对错综复杂的输送线路，小 S 翻出了一叠陈旧的测试记录。参与过测试的小 T 指出，由于当年的测试是顺着电网层级逐级开展的，记录中涉及的节点集合呈现出规整的嵌套关系：任意两份记录涉及的节点集合，要么完全独立，要么存在严格的包含关系。

盲目试错不仅耗时且风险极高。为了尽快配置好供电网络，小 T 和小 S 需要预先还原出每次测试中，将所有节点接入主模块的方案数。

【题目描述】

供电网络包含 n 个节点，节点之间通过若干条双向输送线路相连，构成一张无向图。

在配置网络时，所有节点将被分配至两个独立的主供电模块中。对于节点 i ($1 \leq i \leq n$)，定义其**同模块邻点数** d_i 为：在节点 i 所接入的供电模块内，与其有直接线路连接的节点数量。

小 S 共找出了 q 次测试的记录，每次测试的记录通过一个长度为 n 的字符串 s 来表示。对于第 i ($1 \leq i \leq q$) 个节点：

- 若 $s_i = 0$ ，则要求在该配置下节点 i 的同模块邻点数 d_i 为偶数；
- 若 $s_i = 1$ ，则要求在该配置下节点 i 的同模块邻点数 d_i 为奇数；
- 若 $s_i = ?$ ，则记录中不涉及节点 i ，即对节点 i 的同模块邻点数奇偶性不作要求。

小 T 指出，记录中涉及的节点集合呈现出规整的嵌套关系。具体而言，设第 i ($1 \leq i \leq q$) 份测试涉及的节点集合为 S_i （即对应字符串中所有不为 ? 的位置构成的集合），则对于任意两份不同的测试记录 i, j ($1 \leq i < j \leq q$)， S_i 和 S_j 必定满足以下三种关系之一： $S_i \subseteq S_j$ ， $S_j \subseteq S_i$ ，或 $S_i \cap S_j = \emptyset$ 。

为了尽快配置好供电网络，你需要协助小 T 和小 S 求出，每次测试中将所有节点接入两个主模块的本质不同方案数。两个方案被视为不同，当且仅当存在至少一个节点，在两套方案中被接入了不同的主模块。由于答案可能很大，你只需要求出其对 $10^9 + 7$ 取模后的结果。

【输入格式】

输入的第一行包含两个正整数 n, q ($1 \leq n, q \leq 3 \times 10^3$)。

接下来 n 行，每行包含一个长度为 n 的 01 字符串，其中第 i ($1 \leq i \leq n$) 行的第 j ($1 \leq j \leq n$) 位表示节点 i 与节点 j 间是否存在输送线路，若存在则为 1，否则为 0。

接下来 q 行，每行包含一个长度为 n 的字符串 s ，表示一次测试的记录。

【输出格式】

输出 q 行，每行一个非负整数，表示该次测试中将所有节点接入两个主模块的本质不同方案数对 $10^9 + 7$ 取模后的结果。

【样例 1 输入】

```
1 3 2
2 010
3 100
4 000
5 1?0
6 010
```

【样例 1 输出】

```
1 4
2 0
```

【样例 1 解释】

对于第一次测试，共有以下四种接入方案：

1. 将所有节点接入第一个主模块；
2. 将所有节点接入第二个主模块；
3. 将节点 1,2 接入第一个主模块，将节点 3 接入第二个主模块；
4. 将节点 1,2 接入第二个主模块，将节点 3 接入第一个主模块。

【样例 2 输入】

```
1 6 5
2 000010
3 000001
4 000000
5 000001
```

```
6 100000
7 010100
8 ?11?0?
9 ??????
10 ?10?1?
11 ??0?0?
12 ?01?01
```

【样例 2 输出】

```
1 0
2 64
3 16
4 32
5 0
```

此页刻意留白

K. 流光解密 / Aura

这是一道通信题。

请务必在每次输出后刷新标准输出缓冲区。各编程语言的刷新方式如下：

- 对于 C/C++，请使用 `fflush(stdout)` 或 `cout.flush()`；
- 对于 Java，请使用 `System.out.flush()`；
- 对于 Python，请使用 `sys.stdout.flush()`。

【题目背景】

供电网络修复完毕，全息投影仪终于顺利启动。随着夜色渐深，半空中的全息投影愈发缤纷绚丽。一切就绪后，小 T 和小 S 正式拉开了夜间活动的帷幕，邀请大家结伴，共同参与这场精心筹备、考验双人合作默契的光影游戏“流光解密”。

广场中央的光束交织在一起，缓缓汇聚成一棵全息光树。光树由若干个悬浮的光团结点与连接它们的流光线路组成，呈现出纯粹的树形结构。游戏开始时，所有线路均未点亮，挑战者无法观测到任何未点亮的线路。执行系统操作的 Karuha 会率先向驻守在控制台前的一名挑战者揭示一个神秘数字。随后，各条流光线路会逐一亮起，该挑战者必须在每条线路浮现的瞬间，决定其流动的方向；而站在舞台另一端的搭档，则需仅凭最终形成的有向树结构，准确推断出这个神秘数字。

作为庆典参与者，Neri 和 Noir 决定配合完成这项挑战。

【题目描述】

在这项挑战中，Neri 将驻守在控制台前。掌控系统的 Karuha 首先会向她给出两个正整数 n, s ($1 \leq s \leq 2^{n-1}$)，分别表示全息光树包含的光团数量与神秘数字。

随后，Karuha 将依次点亮 $n - 1$ 条流光线路，Neri 必须在每条线路亮起时，立即为其指定流动的方向。

对于站在主舞台另一端的 Noir，她将观察到整棵全息光树充满流光的最终形态。她需要根据这些线路的流动方向，推断出 Karuha 传递给 Neri 的神秘数字 s 。

为了赢得这场光影挑战，Neri 和 Noir 需要提前制定一套策略，以确保在这个考验双人默契的游戏中顺利通关。

【实现细节】

你的程序将会独立运行两次，其中第一次将扮演 Neri 决定每条线路流动的方向，第二次将扮演 Noir 根据全息光树的最终形态推断出神秘数字。交互器将扮演 Karuha，向你的程序传递信息。

在第一次运行中，你的程序会首先收到全息光树包含的光团数量 n 与神秘数字 s ，然后依次收到 $n - 1$ 条点亮的流光线路。你的程序需要立即向交互库输出为其指定的流

动的方向，从而向第二次运行传递信息。

在第二次运行中，你的程序会从交互器收到来自第一次运行的信息，即全息光树上所有流光线路的流动方向，然后需要根据这些信息推断出 Karuha 传递给 Neri 的神秘数字 s 。

【输入输出格式】

每个测试点中包含多组测试数据。输入的第一行包含两个正整数 T, r ($1 \leq T \leq 10^4, r \in \{1, 2\}$)，表示数据组数与阶段编号。对于每组测试数据：

- 若 $r = 1$ ，则
 - 第一行输入两个正整数 n, s ($3 \leq n \leq 30, 1 \leq s \leq 2^{n-1}$)，分别表示全息光树包含的光团数量与神秘数字；
 - 接下来将会依次点亮 $n - 1$ 条流光线路。每次点亮一条流光线路时，输入两个正整数 u, v ($1 \leq u < v \leq n$)，表示流光线路连接的两个光团的编号，你需要**立即**输出一行两个正整数 u, v 或 v, u ，表示流动方向为从 u 到 v 或从 v 到 u 。
 - **注意：**
 - * 对于每条流光线路，你**必须**输出该条流光线路的流动方向并刷新标准输出缓冲区后，才能继续输入下一条点亮的流光线路。
- 若 $r = 2$ ，则
 - 第一行输入一个正整数 n ($3 \leq n \leq 30$)，表示全息光树包含的光团数量；
 - 接下来 $n - 1$ 行，每行输入两个正整数 u, v ($1 \leq u, v \leq n$)，表示一条从光团 u 流向光团 v 的流光线路。
 - 你需要**立即**输出一行一个正整数，表示推断出的神秘数字 s 。
 - **注意：**
 - * 单个测试点内测试数据输入的先后顺序相较第一次运行时**可能不同**。
 - * 对同一组测试数据，输入所有流光线路的顺序在两次运行中**可能不一致**，但所有光团的编号保持不变。
 - * 对于每一组测试数据，你**必须**输出推断出的神秘数字 s 并刷新标准输出缓冲区后，才能继续输入下一组测试数据。

【样例】

第一次运行时，输入如下：

```
1 2 1
2 7 21
3 1 6
```

```
4 3 5
5 2 4
6 3 4
7 1 2
8 6 7
9 9 59
10 1 2
11 1 4
12 3 4
13 3 5
14 3 8
15 5 6
16 6 7
17 8 9
```

Neri 定向的结果如下：

```
1 6 1
2 3 5
3 4 2
4 3 4
5 1 2
6 7 6
7
8 1 2
9 1 4
10 3 4
11 3 5
12 8 3
13 5 6
14 6 7
15 9 8
```

注意：上述空行仅为方便理解两组测试数据之间的分隔，实际程序输出时无需打印多余的空行。

两组测试数据的所有流光线路定向后，全息光树的形态分别如下：

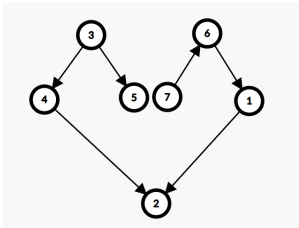


图 1: 第一组测试数据

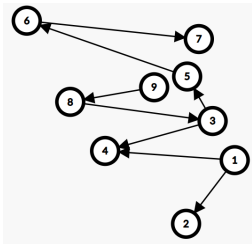


图 2: 第二组测试数据

根据上述输出，第二次运行时，一种可能的输入如下：

1	2	2
2	9	
3	9	8
4	1	2
5	5	6
6	6	7
7	1	4
8	3	5
9	8	3
10	3	4
11	7	
12	1	2
13	4	2
14	6	1
15	3	5
16	3	4
17	7	6

通过心灵感应，Noir 推断出第一组测试数据中 $s = 59$ ，第二组测试数据中 $s = 21$ ，因此输出：

1	59
2	21

【测试程序方式】

下发的脚本文件 `treedir_testing_tool.py` 可以协助你进行本地测试以验证程序的正确性，并打印详细的交互过程。测试时，请将该脚本与你编译生成的可执行文件置于同一目录下，随后在终端内运行以下命令：

```
1 python3 treedir_testing_tool.py [--quiet] <data_file>
   <program> <arguments>
```

其中：

- `-q`, `--quiet` 为可选参数，使用该参数后，测试脚本将不再打印详细的交互过程。
- `data_file` 为提供给测试脚本的输入数据文件路径。该文件需满足以下格式：
 - 第一行包含两个非负整数 $T, seed$ ，分别代表测试数据组数，与用于打乱测试数据顺序及树内连边顺序的随机数种子。若指定 $seed = 0$ ，则表示不进行打乱。
 - 每组数据的格式与前文描述的“第一次运行”的输入格式完全相同。
- `program` 为你编译生成的可执行文件路径。
- `arguments` 为传递给该可执行文件的额外命令行参数。

关于测试脚本的更多实现细节，请直接查阅其源代码。

注意：

1. 测试脚本与实际评测时的交互库实现**不完全相同**，本地测试结果不具有终断性，仅供调试阶段参考；
2. 测试脚本仅会对 `data_file` 中的数据进行基础的格式校验，**不会检查**其是否合法地满足了题目限制（例如，不会检查当 s 是否满足 $1 \leq s \leq 2^{n-1}$ 的限制，也不会判断所有流光线路能否正确构成一棵树）；
3. 测试脚本**不会监控**程序的时间与空间消耗，无法用于测试是否符合题目的时空限制。

此页刻意留白

L. 星图重绘 / Redrawing

【题目背景】

随着流光解密挑战的圆满完成，全息光树被完全点亮，盛大的十周年庆典也悄然步入尾声。作为庆典的最后环节，小 T 和小 S 邀请大家来到了一块描绘着漫天星图的数字画板前。

岁月流转，过往绘就的星座连线已随着时间悄然淡化，此时的画板又回到了仅有孤立星辰散布的模样。小 S 递给现场的大家特制的星图笔，希望大家能重新补全这份星图。为了使重绘的纪念星图呈现出极致的美感，连结星辰时必须保持结构的平衡。

伴随着悠扬的旋律，大家纷纷上前，希望能在这片星海中点亮尽可能多的星座，为这场十周年纪念晚会留下最璀璨的收官之作。

【题目描述】

画板上共有 n 颗星辰，可将其视为二维平面上的点。第 i ($1 \leq i \leq n$) 颗星辰的坐标为 (x_i, y_i) 。已知所有星辰的横坐标两两不同，纵坐标也两两不同。

每次绘制星座时，你需要从这 n 颗星辰中选取三颗，将它们两两连接构成一个三角形。为了体现平衡的美感，该三角形必须满足特定的边界要求：存在一个四条边均平行于坐标轴的矩形，使得该三角形的三个顶点恰好都落在这个矩形的边界上。同时，为了维持星图的清晰结构，所有连结出的三角形的内部区域（不包含顶点与边界）必须两两互不相交。

请你计算出最多能成功绘制多少个星座，并给出一组具体的绘制方案。

【输入格式】

本题包含多组测试数据。输入的第一行包含一个正整数 t ($1 \leq t \leq 2 \times 10^4$)，表示数据的组数。

接下来的每组数据，第一行包含一个正整数 n ($3 \leq n \leq 2 \times 10^5$)。

接下来 n 行，第 i ($1 \leq i \leq n$) 行包含两个整数 x_i, y_i ($|x_i|, |y_i| \leq 10^9$)，表示第 i 颗星辰的坐标。保证所有 x_i 两两不同，所有 y_i 两两不同。

【输出格式】

对每组数据，第一行输出一个非负整数 m ，表示最多能绘制的星座数量。

接下来 m 行，每行输出三个互不相同的正整数 x, y, z ($1 \leq x, y, z \leq n$)，表示构成一个星座的三颗星辰。

【样例 1 输入】

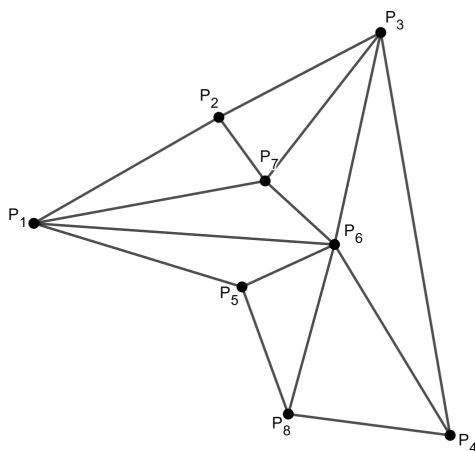
```
1 2
2 8
3 -10 1
4 -2 6
5 5 10
6 8 -9
7 -1 -2
8 3 0
9 0 3
10 1 -8
11 8
12 8 8
13 -5 3
14 -4 1
15 5 7
16 10 10
17 -3 5
18 -8 -10
19 -7 -1
```

【样例 1 输出】

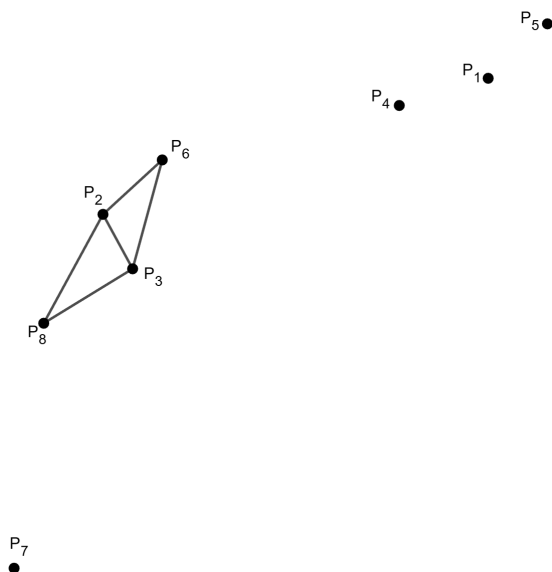
```
1 8
2 6 5 8
3 6 8 4
4 1 6 5
5 7 1 6
6 2 7 1
7 3 2 7
8 3 7 6
9 3 6 4
10 2
11 2 3 8
12 6 2 3
```

【样例 1 解释】

对于第一组测试数据，样例输出中绘制的星图如下图所示：



对于第二组测试数据，样例输出中绘制的星图如下图所示：



此页刻意留白

M. 庆典留念 / Souvenir

【题目描述】

至此，THUPC 十周年庆典即将结束。小 T 和小 S 站在主会场的出口处，为每一位满载而归的同学们准备了精美的留念礼品。

十年的岁月沉淀了无数闪光的瞬间，而今晚的相聚同样令人难忘。经历了这一系列智力与欢乐的碰撞，大家心中想必都有了自己最钟意的活动。为此，小 T 和小 S 将根据大家的不同选择，分别赠上对应主题的十周年定制纪念品。

请你从本次庆典的所有活动项目中，挑选出你最喜欢的一个，从而获取这份专属的纪念品吧！

【输入格式】

本题没有输入。

【输出格式】

输出一行一个仅包含英文字母（不区分大小写）的字符串，表示你最喜欢的活动项目。

【样例 1 输入】

```
1 THUPC 2026
```

【样例 1 输出】

```
1 GOOD LUCK & HAVE FUN!
```

【结语】

感谢不远千里奔赴而来的各位选手，
感谢辛勤付出的全体工作人员，
感谢慷慨解囊的各位赞助商，
感谢一直以来帮助与指导算协的各位老师，
感谢过去十年间与清华大学学生算法协会相遇的每一份缘分。
第十年，我们还没倒闭!!!!!!!!!!!!

此页刻意留白

此页刻意留白