

2026 年秋季华东理工大学程序设计竞赛

正式赛 试题册

2026 年 9 月 29 日 13 : 00 ~ 18 : 00

试题列表

题号	试题名称	时空限制	注释
A	纽约提喻法	2s/1GB	
B	AIMC 实验室	2s/1GB	
C	一分为二 2	2s/1GB	
D	位距!?	2s/1GB	
E	根除	4s/1GB	
F	凸包	2s/1GB	
G	星之彼端	8s/1GB	通信题
H	一般攻击魔法	2s/1GB	
I	天外来物	1s/1GB	
J	时间谜题 2	1s/1GB	
K	永世沙漏	2s/1GB	
L	净化行动 3	2s/1GB	
M	魔女审判	6s/1GB	

试题册共包含 13 道试题和 21 页。

若您的试题册缺页请立刻联系志愿者。

请务必认真阅读位于比赛文件附件中的赛事规则和技术规则。

赛事开始前请勿翻阅试题册，亦请勿触碰鼠标和键盘。

请注意黑白打印的试题册可能会丢失部分需要颜色展现的信息。

写在题目描述前的斜体文字是 *Feature Text* (风味文本)，和题目无关，包括这句话。

试题 A 纽约提喻法

运行时间限制 2 seconds
内存使用限制 1 Gigabyte
读入文件 标准输入
输出文件 标准输出

*And they say there is no fate, but there is, it's what you create.
And even though the world goes on for eons and eons...
...you are only here for a fraction of a fraction of a second.*

题目描述

凯顿对“命运”有了自己的理解，他不再借助戏剧，而想用一串字符来提喻自己的人生。然而，当他尝试构建这个提喻时，为了追求纯粹、忠于自我的原创表达，他再度陷入了自我指涉的无限循环。于是这一次，凯顿将选择权交给了你，由你来描绘他的人生。

凯顿的提喻是一个长度为 L 的字符串，记为 T 。称 T 是合法的，当且仅当它同时满足以下三个条件：

1. T 仅由小写英文字母组成。
2. 对于每个 $i = 1, 2, \dots, K$ ，字符串 T 的第 p_i 个字符必须等于给定的字符 c_i 。这里 p_i 是位置下标，满足 $1 \leq p_i \leq L$ ， c_i 为小写字母。
3. 对于每个 $i = 1, 2, \dots, N$ ，给定的字符串 s_i 不能作为 T 的连续子串出现。

现在给定 L, K 组位置约束 (p_i, c_i) 以及 N 个禁止子串 s_i ，请你计算所有合法字符串 T 的总数，并将结果对 998 244 353 取模后输出。

输入格式

第一行输入三个整数 L, K, N ($1 \leq L \leq 10^{18}$, $0 \leq K \leq 500$, $0 \leq N \leq 50$)，分别表示字符串长度、位置约束数量和禁止子串数量。

随后 K 行，每行输入一个整数 p_i 和一个小写字母 c_i ($1 \leq p_i \leq L$)，表示一个位置约束。保证所有 p_i 互不相同。

随后 N 行，每行输入一个非空字符串 s_i ，表示一个禁止子串。

保证所有 s_i 的长度之和不超过 100 (即 $\sum_{i=1}^N |s_i| \leq 100$)。

输出格式

对于每组测试数据，输出一行一个整数，表示合法字符串的总数对 998 244 353 取模后的结果。

样例 1

standard input	standard output
3 1 1	650
2 b	
ab	

样例 2

standard input	standard output
10 2 2	113392371
3 a	
5 c	
abc	
dcabe	

样例 1 解释

字符串长度为 3，第 2 个字符必须是 **b**，且不能包含子串 **ab**。合法的字符串形如 **?b?**，其中第 1 个位置可以是除 **a** 以外的任意小写字母，共 $25 \times 26 = 650$ 种。

试题 B AIMC 实验室

运行时间限制 2 seconds
内存使用限制 1 Gigabyte
读入文件 标准输入
输出文件 标准输出

街灯璀璨，可我日渐透明。

题目描述

你所在的 AIMC 实验室研究出了一种全新的三维量子矩阵！

给定一个大小为 $N \times M \times H$ 的三维空间网格，坐标范围为 $1 \leq x \leq N, 1 \leq y \leq M, 1 \leq z \leq H$ 。在其中每个整数格点上都有一个量子比特，记量子比特 u 的坐标为 (x_u, y_u, z_u) 。

对于任意两个量子比特 u 和 v ，定义它们之间的平方空间距离为：

$$d^2(u, v) = (x_u - x_v)^2 + (y_u - y_v)^2 + (z_u - z_v)^2.$$

系统存在 8 个形状和网格一样同为 $N \times M \times H$ 的能量矩阵 $A_{D,c}$ 其中两个整数 $0 \leq D \leq 3, 0 \leq c \leq 1$ 。

对于网格中的每一个量子比特，你可以将其初始化为两种基态之一： $|0\rangle$ 态或 $|1\rangle$ 态，此时其基态为 $c_P = 0$ 或 $c_P = 1$ 。

在初始化为基态后，系统总能量计算规则如下：

- 对于每个整数 $0 \leq D \leq 3, 0 \leq c \leq 1$ ，量子比特 P 会获得 $A_{D,c}(x_P, y_P, z_P)$ 的能量，当且仅当：
 - P 的基态满足 $c_P = c$ 。
 - 对于任意量子比特 Q 满足其 $d^2(P, Q) = D$ ，有其 $c_P + c_Q + D$ 为偶数。

作为 AIMC 实验室的首席科学家，请你为网格中的每个量子比特初始化为一个基态，使得整个系统获得的总能量最大化，并输出这个最大能量值。

输入格式

第一行输入三个整数 N, M, H ($N \times M \times H \leq 100$)，表示三维网格的大小。

接下来依 (D, c) 的字典序给出 8 个能量观测矩阵 $A_{D,c}$ 的数据，

对于每个矩阵输入，包含 N 个区块。每个区块包含 M 行，每行 H 个整数。区块之间用空行隔开。

第 x 个区块的第 y 行第 z 个数即代表对应坐标 (x, y, z) 的能量值，保证所有能量值 $A_{D,c}(x_P, y_P, z_P) \in [0, 10^9]$ 。

输出格式

输出一行一个整数，表示系统可以获得的最大总能量值。

样例 1

standard input	standard output
1 1 1	2265
3	
5	
100	
200	
1000	
2000	
50	
60	

样例 1 解释

整个网格仅有一个量子比特 $(1, 1, 1)$ 。

将该量子比特初始化为 $|1\rangle$ 时获得 $A_{0,1}(1, 1, 1) + A_{1,1}(1, 1, 1) + A_{2,1}(1, 1, 1) + A_{3,1}(1, 1, 1) = 5 + 200 + 2000 + 60 = 2265$ ，为最优。

样例 2

standard input	standard output
2 2 2	88
1 1	
1 1	
1 1	
1 1	
1 1	
1 1	
0 0	
0 0	
0 0	
0 0	
0 0	
0 0	
0 0	
0 0	
0 0	
10 10	
10 10	
10 10	
10 10	
0 0	
0 0	
0 0	
0 0	
0 0	
0 0	
0 0	
0 0	
0 0	
0 0	
0 0	

样例 2 解释

将 $2 \times 2 \times 2$ 中所有 8 个量子比特统一初始化为 $|0\rangle$ 。

- 每个量子比特从 $A_{0,0}$ 各获得 1 的能量，合计 8。
- 每个量子比特距离平方距离为 2 的量子比特也全部为 $|0\rangle$ ，从 $A_{2,0}$ 各获得 10，合计 80。

总能量 $8 + 80 = 88$ 。可以证明这是最大值。

试题 C 一分为二 2

运行时间限制 2 seconds
内存使用限制 1 Gigabyte
读入文件 标准输入
输出文件 标准输出

别在昨日里频频回首，说什么将寂寞对半分担。

题目描述

给定两个长度为 n 的数组 a 和 b 。
你可以任选 $k \in [0, n]$ ，将 a 的前 k 个元素和 b 的后 $n - k$ 个元素拼接成一个长度为 n 的新数组。
请从新数组中选出恰好 m 个互不相邻的元素。记其中最小的元素为 p ，求 p 的最大值。

输入格式

第一行输入一个整数 T ($1 \leq T \leq 10^4$)，表示测试数据的数量。
每组测试数据第一行输入两个整数 n 和 m ($1 \leq m \leq \lceil \frac{n}{2} \rceil \leq n \leq 2 \times 10^5$)，分别表示数组长度和需要选择的元素数量。
第二行输入 n 个整数 $a_1, a_2, \dots, a_n$ ($-10^9 \leq a_i \leq 10^9$)，表示数组 a 。
第三行输入 n 个整数 $b_1, b_2, \dots, b_n$ ($-10^9 \leq b_i \leq 10^9$)，表示数组 b 。
保证所有测试数据的 $\sum n \leq 2 \times 10^5$ 。

输出格式

对于每组测试数据，输出一行一个整数，表示 p 的最大值。

样例 1

standard input	standard output
2	5
5 2	-4
5 4 3 2 1	
1 2 3 4 5	
9 5	
-1 1 -4 -5 -1 4 -1 -9 1	
-9 -9 -8 2 4 -4 -3 -5 3	

样例 1 解释

对于第 1 组测试数据，当 $k = 4$ 时，新数组为 $[5, 4, 3, 2, 5]$ ，选择第 1, 5 个元素，最小元素为 5，可以证明这是最大值。

试题 D 位距!?

运行时间限制 2 seconds
内存使用限制 1 Gigabyte
读入文件 标准输入
输出文件 标准输出

最后跳支舞吧。
在魔都的远郊，
在化工强校的信息楼，
在这个光怪陆离的
革命前夜。

题目描述

请在欧氏平面上构造一个 n 个点的点集，使得：

- 每个点的横坐标和纵坐标绝对值分别不超过 10^5 ；
- 每个点的横坐标和纵坐标分别乘上 10^{13} 是一个整数。
- 任意两点之间的距离不小于 10^{-4} ；
- 至少存在 $\lfloor \frac{1}{2}n \log_2 n - n \rfloor$ 个不同的无序点对距离在 $[1 - 10^{-9}, 1 + 10^{-9}]$ 之间。

输入格式

第一行输入一个整数 T ($1 \leq T \leq 100$)，表示测试数据的数量。
每组测试数据输入一行一个整数 n ($2 \leq n \leq 5000$)，表示需要选定的点的数量。
保证所有测试数据的 $\sum n \leq 5000$ 。

输出格式

对于每组测试数据，输出 n 行，每行输出两个整数 X_i, Y_i ($-10^{18} \leq X_i, Y_i \leq 10^{18}$)，表示第 i 个点的坐标实际为 $(\frac{X_i}{10^{13}}, \frac{Y_i}{10^{13}})$ 。
你的输出将被视为正确，当且仅当它符合上述输出格式要求及题目描述中的四个条件。
如果有多个合法的答案，则你可以输出其中任意一个。

样例 1

standard input	standard output
1	0 0
3	10000000000000 0
	5000000000000 8660254038000

样例 1 解释

对于第 1 组测试数据，实际点坐标为：

$$(x_1, y_1) = (0, 0), (x_2, y_2) = (1, 0), (x_3, y_3) = (0.5, 0.8660254038)$$

前三个条件不难验证，对于第三个条件，有：

$$d_{12} = 1, d_{13} = d_{23} = 1.000000000013476527219909191607 \dots \approx 1 + 1.3476 \times 10^{-11}$$

因此每个点对之间的距离都在 1 的误差范围内，这是符合第四个条件的。

试题 E 根除

运行时间限制 4 seconds
内存使用限制 1 Gigabyte
读入文件 标准输入
输出文件 标准输出

我们终于等来了黎明，可这座城的根除了记得黑夜，似乎什么也没能留下。

题目描述

我们通常会把一个较大的数字的开根分解为最简的根式，例如 $\sqrt{432} = 12\sqrt{3}$ 。
在 $\mathbb{N}$ 上定义数论函数 $f(n) = \min\{a \mid n = ab^2, a, b \in \mathbb{N}\}$ ，即函数值为满足等式的最小自然数 a ，可以发现这样的 a 总是存在的。
给定 q 次询问，每次询问给定 l, r ，请分别求出 $\sum_{i=l}^r f(i)$ 的答案在模 998 244 353 下的值。

输入格式

第一行输入一个整数 q ($1 \leq q \leq 500$)，表示询问的数量。
随后输入 q 行两个整数 l 和 r ($1 \leq l \leq r \leq 10^{12}$)，表示一次询问的区间。

输出格式

对于每次询问，输出一行一个整数，表示 $\sum_{i=l}^r f(i)$ 在模 998 244 353 下的值。

样例 1

standard input	standard output
4	3
432 432	38
1 10	66
10 15	651260790
1 1000000000	

样例 1 解释

对于第 2 次和第 3 次询问，对应的 f 值为：

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$f(n)$	1	2	3	1	5	6	7	2	1	10	11	3	13	14	15

其中第 3 次询问即为

$$\sum_{i=10}^{15} f(i) = f(10) + f(11) + f(12) + f(13) + f(14) + f(15) = 10 + 11 + 3 + 13 + 14 + 15 = 66.$$

试题 **F** 凸包

运行时间限制 2 seconds
内存使用限制 1 Gigabyte
读入文件 标准输入
输出文件 标准输出

我知道既不是凸包也不是凹包的我为什么要哭的。因为我其实是一只豆包。

题目描述

给定 n 个一元二次不等式，每个不等式的形式为：

$$a_i x^2 + b_i x + c_i > 0$$

对任意 $1 \leq i \leq n$ ，保证 $a_i < 0$ 。

请你判断是否存在整数 x ，使得这 n 个不等式同时成立。

输入格式

第一行输入一个整数 T ($1 \leq T \leq 10^4$)，表示测试数据的数量。

每组测试数据第一行输入一个整数 n ($1 \leq n \leq 2 \times 10^5$)，表示不等式的个数。

随后 n 行，每行输入三个整数 a_i, b_i, c_i ($-10^5 \leq a_i \leq -1, -10^5 \leq b_i, c_i \leq 10^5$)，表示第 i 个不等式的三个系数。

保证所有测试数据的 $\sum n \leq 2 \times 10^5$ 。

输出格式

对于每组测试数据，若存在整数 x 同时满足所有不等式，输出一行 YES，否则输出一行 NO。

注意评测时不会区分 YES 和 NO 的大小写。换言之，当答案是肯定的时候输出 yes、YES、Yes、YeS 等都会被认为是正确的。

样例 1

standard input	standard output
5	NO
2	YES
-1 0 4	NO
-1 4 -3	YES
3	NO
-1 100000 0	
-1 202 10000	
-2 400 19998	
2	
-1 2 0	
-1 0 -1	
2	
-100000 0 100000	
-1 0 4	
2	
-1 2 0	
-1 4 -3	

样例 1 解释

对于第 4 组测试数据，当 $x = 0$ 时，两个不等式均成立，因此存在整数解。

试题 G 星之彼端

运行时间限制 8 seconds
内存使用限制 1 Gigabyte
读入文件 标准输入
输出文件 标准输出

今夜天空深远宁静，我们望的是同一颗星星吗？

这是一道通信题，请注意本题不同寻常的数据范围约束。

题目描述

那是整个 ECUST 都早已遗忘的久远过去……

在那个无人记得的年代，竹林和千外并肩战斗，共同生活，一起仰望同一片星空。后来，语言被风沙掩埋，道路被海洋截断，连名字都从史书里褪色。唯一没有被带走的，是头顶那片沉默而辽阔的星图。它公开、闪耀、亘古不变，却没有人知道其中哪几颗星曾被他们悄悄连成一句话。

本题共有 20 个测试点。

给定一个无向图 $G = (V, E)$ ，其中 $|V| = 2000$, $|E| = 43\,600$ 。图 G 不包含重边和自环，并且保证由以下方式生成：初始时，图 G 包含 $|V|$ 个点且没有任何边。随后，生成器将重复执行以下操作 $|E|$ 次：在当前所有尚未被边连接的点对中，等概率地随机选取一对点，并在它们之间添加一条无向边。

你的程序将被运行两次：

第一次运行

在第一次运行时，你将扮演竹林。你需要在图 G 中选择 n 个诱导四元环并输出。这些诱导四元环可以拥有共同的顶点。

- **诱导四元环的定义**：选定图 G 中的 4 个点，这 4 个点之间恰好存在 4 条边，且构成一个简单的环（即内部不能有对角线相连）。

第二次运行

在第二次运行时，你将扮演千外。评测机将获取你在第一次运行中输出的所有四元环。你需要回答 q 次询问。对于每一个询问，评测机会挑选一个四元环，任选其中一个点并删除，然后输出剩余三个点的编号。你需要回答这三个点来自于第一次运行时输出的第几个四元环。

这三个点可能同时来自于你挑选的多个多元环，但你必须回答评测机挑选的那一个。

输入格式

第一次运行

第一行包含一个字符串 `send`。

第二行包含一个整数 n ($n = 200\,000$)，表示需要输出的诱导四元环数量。

第三行包含两个整数 $|V|$ 和 $|E|$ 。

接下来 $|E|$ 行，每行包含两个整数 u, v ，表示顶点 u 和 v 之间有一条无向边 ($1 \leq u, v \leq |V|$)。

第二次运行

第一行包含一个字符串 `receive`。

第二行包含两个整数 $|V|$ 和 $|E|$ 。

接下来 $|E|$ 行，每行包含两个整数 u, v ，表示顶点 u 和 v 之间有一条无向边 ($1 \leq u, v \leq |V|$)。

第 $|E| + 3$ 行包含一个整数 q ($q = 4n$)，表示接收到的顶点集合数量。

接下来的 q 行，每行包含 3 个整数，代表评测机从你第一次运行输出的某个四元环中删除一个顶点后得到的顶点集合。

输出格式

第一次运行

输出 n 行。第 i 行输出 4 个整数 v_1, v_2, v_3, v_4 ，代表第 i 个诱导四元环的四个顶点编号。

第二次运行

输出 q 行。对于每一组接收到的顶点集合，输出一个整数 i ($1 \leq i \leq n$)，表示该集合属于第 i 个四元环。

样例 1

第一次运行

standard input	standard output
send	1 174 613 1323
200000	1 955 1683 1755
2000 43600	...
29 1973	1390 1615 1683 1777
228 1127	
...	
92 573	

第二次运行

standard input	standard output
receive	17713
2000 43600	191947
29 1973	...
228 1127	118231
...	
92 573	
800000	
46 1297 1949	
1056 1275 1454	
...	
387 1289 1533	

样例 1 解释

由于完整数据过大，题面省略了样例的部分内容。你可以在附件中查看第一次运行的完整输入。

试题 H 一般攻击魔法

运行时间限制 2 seconds
内存使用限制 1 Gigabyte
读入文件 标准输入
输出文件 标准输出

“Zoltraak.”

题目描述

在一个 $n \times m$ 的二维网格图中，网格中的每个格子可以处于以下三种状态之一：

1. /（正斜杠镜子）：双面反光。光线在此发生 90° 偏折，即连通（上方与左方）以及（下方与右方）。
2. \（反斜杠镜子）：双面反光。光线在此发生 90° 偏折，即连通（上方与右方）以及（下方与左方）。
3. .（空格子）：无障碍物。光线在此不发生偏折，即连通（上方与下方）以及（左方与右方）。

初始时，网格中的部分格子已经固定放入了 / 或 \ 镜子。其余格子目前为空。对于每一个初始为空的格子，你可以选择将其保留为空，或者将其替换为 / 镜子，或者将其替换为 \ 镜子。

整个网格的外侧由 $2(n+m)$ 条单位长度的线段包围。光线只能从这些线段的中点垂直射入或射出。

现给定一束光线的初始射入位置以及目标射出位置，请问是否存在一种合法的网格填充方案，使得光线从指定位置射入网格后，经过若干次直线传播与镜面反射，最终恰好在指定位置射出？

如果存在合法方案，请构造一个填充后的网格，使得光线从指定位置射入并从目标位置射出的过程中，经过的格子数最少。每进入一个格子计数一次；若多次经过同一个格子，则重复计数。若有多种最优方案，输出任意一种即可。否则，请报告无解。

输入格式

第一行输入一个整数 T ($1 \leq T \leq 100$)，表示测试数据的数量。

每组测试数据第一行输入两个整数 n 和 m ($1 \leq n \times m \leq 2 \times 10^5$)，表示网格的行数和列数。

随后 n 行，每行输入一个长度为 m 的字符串，表示网格的初始状态。其中 / 表示正斜杠镜子，\ 表示反斜杠镜子，. 表示空格子。

随后一行包含四个用空格隔开的参数 d_1, p_1, d_2, p_2 。前两个参数描述初始射入位置，后两个参数描述目标射出位置。其中：

- d_1, d_2 为字符，属于 U, D, L, R 之一，分别表示网格的上、下、左、右边界。
- p_1, p_2 为正整数，表示光线从该边界的第 p 个格子的外边缘中点垂直射入或射出。若边界为 U 或 D，则 p 表示列号 ($1 \leq p \leq m$)；若边界为 L 或 R，则 p 表示行号 ($1 \leq p \leq n$)。

保证 $(d_1, p_1) \neq (d_2, p_2)$ 。

保证所有测试数据的 $\sum(n \times m) \leq 2 \times 10^5$ 。

输出格式

对于每组测试数据，若存在合法方案，输出 n 行，每行输出一个长度为 m 的字符串，表示使光线经过的格子数最少的填充方案。若有多种最优方案，输出任意一种即可。否则，输出一行 IMPOSSIBLE。

请注意，在绝大多数编程语言中，为了输出 \，你需要使用 \\ 来进行转义。

样例 1

standard input

```
3
2 2
..
..
L 1 U 2
4 4
^ ^
^ /.
....
./..
L 2 L 4
4 4
^ ^
^ /.
.. \
./..
L 2 L 4
```

standard output

```
./
..
^ ^
^ //
.//.
./..
IMPOSSIBLE
```

样例 1 解释

对于第 1 组测试数据，光线从第 1 行的左边界射入（向右），经过 (1,1) 遇到 . 继续向右，经过 (1,2) 遇到 / 向上偏折，从第 2 列的上边界射出（向上）。
该方案经过 2 个格子，达到最小值。

试题 I 天外来物

运行时间限制 1 second
内存使用限制 1 Gigabyte
读入文件 标准输入
输出文件 标准输出

在燥热的季节，一颗流星划过天际，留下光亮的拖尾。
我循着光奔跑的那个夜晚，便是故事的开端。

题目描述

有 n 颗水晶，从左到右编号为 1 到 n 。第 i 颗水晶初始时含有 a_i 单位的能量。你可以按照任意顺序依次采集这 n 颗水晶。
当你采集编号为 i 的水晶时：

- 你获得该水晶当前所含的能量值，并累加到总得分中。
- 如果 $i > 1$ 且编号为 $i - 1$ 的水晶还未被采集，它的能量值会变成其当前能量的相反数。
- 如果 $i < n$ 且编号为 $i + 1$ 的水晶还未被采集，它的能量值会变成其当前能量的相反数。

注意，水晶的编号不会随采集发生变化。
采集完所有 n 颗水晶后，可能的最大总得分是多少？

输入格式

第一行输入一个整数 T ($1 \leq T \leq 10^4$)，表示测试数据的数量。
每组测试数据第一行输入一个整数 n ($1 \leq n \leq 2 \times 10^5$)，表示水晶的数量。
第二行输入 n 个整数 $a_1, a_2, \dots, a_n$ ($-10^9 \leq a_i \leq 10^9$)，表示每颗水晶初始的能量值。
保证所有测试数据的 $\sum n \leq 2 \times 10^5$ 。

输出格式

对于每组测试数据，输出一行一个整数，表示可能的最大总得分。

样例 1

standard input	standard output
2	15
5	47
1 2 3 4 5	
9	
9 -9 8 -2 4 -4 3 -5 3	

样例 1 解释

对于第 1 组测试数据，按照编号 $3 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 4$ 的顺序采集可以使得最大得分达到 15。可以证明这是最大值。

试题 J 时间谜题 2

运行时间限制 1 second
内存使用限制 1 Gigabyte
读入文件 标准输入
输出文件 标准输出

“对无数时间线建立可持久化字典树即可实现对历史的版本控制”，一位青铜龙朋友总是这么告诉我。我对此深信不疑，以至于忘记了自己倒数的心跳。

题目描述

星鱼为了证明他上课从不迟到，坚称自己是在 7:110 踏进了 8:00 开始上课的教室。这启发我们去定义一种“广义时间表示法”。

在广义时间表示法中，时间被记为 $A:B$ (A, B 均为非负整数)。

我们定义广义时间之间存在两种等价关系：

- 对于任意 $B \geq 60$ ， $A:B$ 与 $(A+1):(B-60)$ 是表示同一个时刻的等价时间。
- 对于任意 $A \geq 24$ ， $A:B$ 与 $(A-24):B$ 是表示同一个时刻的等价时间。

我们称一个时间 $h:m$ 是“标准时间”，当且仅当它满足 $0 \leq h < 24$ 且 $0 \leq m < 60$ 。

现在给定一个广义时间，请你求出与它等价的唯一“标准时间”。

输入格式

第一行输入一个整数 T ($1 \leq T \leq 10^5$)，表示测试数据的数量。

每组测试数据输入一行两个整数 A 和 B ($0 \leq A, B \leq 10^9$)，表示一个广义时间 $A:B$ 。

输出格式

对于每组测试数据，输出一行两个整数 h 和 m ，表示与给定广义时间等价的“标准时间” $h:m$ 。

样例 1

standard input	standard output
5	8 50
7 110	1 30
25 30	0 0
0 0	23 59
23 59	7 20
100 200	

样例 1 解释

对于第 1 组测试数据， $7:110 \rightarrow 8:50$ （应用规则 1， $110 = 60 + 50$ ）。

对于第 2 组测试数据， $25:30 \rightarrow 1:30$ （应用规则 2， $25 = 24 + 1$ ）。

对于第 3 组测试数据， $0:0$ 本身就是“标准时间”。

试题 K 永世沙漏

运行时间限制 2 seconds
内存使用限制 1 Gigabyte
读入文件 标准输入
输出文件 标准输出

任季节一再轮转，我却始终没能习惯。

题目描述

在欧氏平面上给定一个顶点按照逆时针顺序给出的多边形 $P = P_0P_1 \cdots P_{n-1}$ 。
记一个角度 α ($0 \leq \alpha < \pi$) 为一个**稳定方向**，当且仅当存在 $c \in \mathbb{R}$ ，使得对于多边形 P （含边界）内的任意一点 X 到直线

$$\cos \alpha x + \sin \alpha y + c = 0$$

的垂线段完全包含在 P （含边界）内，这里 α 是直线的一个法向方向。
请求出满足条件的**稳定方向**的数量，可以证明这个数量是有限的。

输入格式

第一行输入一个整数 T ($1 \leq T \leq 100$)，表示测试数据的数量。
每组测试数据第一行输入一个整数 n ($3 \leq n \leq 10^5$)，表示多边形的顶点数量。
随后 n 行，每行输入两个整数 x_i, y_i ($-10^9 \leq x_i, y_i \leq 10^9$)，表示多边形的第 i 个顶点的坐标，按逆时针顺序给出。
保证单组测试数据中所有顶点两两不同，相邻两条边仅在公共端点处相交，不相邻的两条边没有公共点。
保证所有测试数据的 $\sum n \leq 10^5$ 。

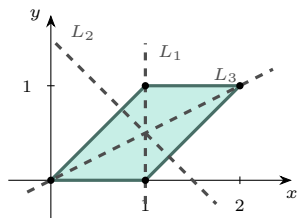
输出格式

对于每组测试数据，输出一行一个整数，表示满足条件的**稳定方向**的数量。

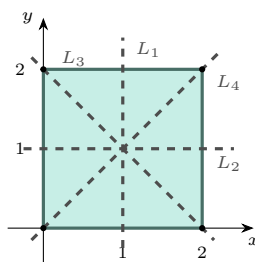
样例 1

standard input	standard output
3	3
4	4
0 0	1
1 0	
2 1	
1 1	
4	
0 0	
2 0	
2 2	
0 2	
6	
0 0	
4 0	
4 1	
1 1	
3 4	
0 4	

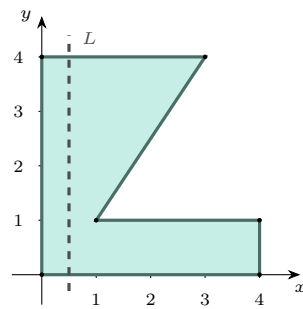
样例 1 解释



第 1 组测试数据的示意图



第 2 组测试数据的示意图



第 3 组测试数据的示意图

第 1 组测试数据的直线和稳定方向对应

$$\begin{aligned} L_1: x - 1 &= 0, & \alpha_1 &= 0, \\ L_2: \frac{\sqrt{2}}{2}x + \frac{\sqrt{2}}{2}y - \frac{3\sqrt{2}}{4} &= 0, & \alpha_2 &= \frac{\pi}{4}, \\ L_3: -\frac{1}{\sqrt{5}}x + \frac{2}{\sqrt{5}}y &= 0, & \alpha_3 &= \pi - \arctan 2. \end{aligned}$$

第 2 组测试数据的直线和稳定方向对应

$$\begin{aligned} L_1: x - 1 &= 0, & \alpha_1 &= 0, \\ L_2: y - 1 &= 0, & \alpha_2 &= \frac{\pi}{2}, \\ L_3: \frac{\sqrt{2}}{2}x + \frac{\sqrt{2}}{2}y - \sqrt{2} &= 0, & \alpha_3 &= \frac{\pi}{4}, \\ L_4: -\frac{\sqrt{2}}{2}x + \frac{\sqrt{2}}{2}y &= 0, & \alpha_4 &= \frac{3\pi}{4}. \end{aligned}$$

第 3 组测试数据的直线和稳定方向对应

$$L: x - \frac{1}{2} = 0, \quad \alpha_1 = 0.$$

注意对应的直线不是唯一确定的。

试题 L 净化行动 3

运行时间限制 2 seconds
内存使用限制 1 Gigabyte
读入文件 标准输入
输出文件 标准输出

除恶务尽——何为「恶」，我们自有裁定。

题目描述

Maddy 仍在清剿 Baddy 的邪恶军团……（这已经是第三次了！）
游戏在一个无限大的平面直角坐标系（可以看作无限大的棋盘）上进行。Baddy 操控了 n 个炮棋子。这些炮分别部署在 $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ 上，且固定不会移动。
在象棋规则中，炮的攻击规则是“隔山打牛”。如果两个炮 A 和 B 处于同一行或同一列，且它们之间有且仅有一个其他炮 C ，那么我们称 A 攻击了 B （同时 B 也攻击了 A ），而 C 就是它们攻击的“炮架”。
每一次，Maddy 可以选择一个当前没有受到任何其他炮攻击的炮，并将其摧毁。被摧毁的炮会被立即移出棋盘，它将不再能攻击其他炮，也不能再作为其他炮的“炮架”。
Maddy 的目标是摧毁所有的 n 个炮。是否存在一种摧毁顺序，使得 Maddy 可以最终清空棋盘？

输入格式

第一行输入一个整数 T ($1 \leq T \leq 10^5$)，表示测试数据的数量。
每组测试数据第一行输入一个整数 n ($1 \leq n \leq 2 \times 10^5$)，表示炮的数量。
随后 n 行，每行输入两个整数 x_i, y_i ($-10^9 \leq x_i, y_i \leq 10^9$)，表示第 i 个炮的位置。
保证所有测试数据的 $\sum n \leq 2 \times 10^5$ ，且单组测试数据中所有炮的位置两两不同。

输出格式

对于每组测试数据，若存在一种合法的摧毁顺序能够清空棋盘，则输出一行 YES，否则输出一行 NO。
注意评测时不会区分 YES 和 NO 的大小写，换言之，当答案是肯定的时候输出 yes、YES、Yes、YeS 等都会被认为是正确的。

样例 1

standard input	standard output
3	YES
3	NO
0 0	NO
0 1	
0 2	
4	
0 0	
1 0	
2 0	
3 0	
8	
2 0	
0 1	
1 1	
2 1	
1 2	
2 2	
3 2	
1 3	

样例 1 解释

对于第 1 组测试数据，三个炮在同一列上。初始时， $(0,0)$ 和 $(0,2)$ 互相攻击（以 $(0,1)$ 为炮架），而 $(0,1)$ 没有受到攻击。因此可以先摧毁 $(0,1)$ ，然后 $(0,0)$ 和 $(0,2)$ 都不再受到攻击，可以按任意顺序摧毁。

对于第 2 组测试数据，四个炮在同一行上。初始时， $(0,0)$ 和 $(2,0)$ 互相攻击（以 $(1,0)$ 为炮架）， $(1,0)$ 和 $(3,0)$ 互相攻击（以 $(2,0)$ 为炮架）。因此 $(0,0)$ 、 $(1,0)$ 、 $(2,0)$ 、 $(3,0)$ 都受到攻击，无法摧毁任何一个，因此无解。

试题 M 魔女审判

运行时间限制 6 seconds
内存使用限制 1 Gigabyte
读入文件 标准输入
输出文件 标准输出

我宁可成为异端，也不愿背叛良知。

题目描述

你拿着的三叉戟再次击中了村民，但这一次，当审判结束，你却发现自己已经找不到它了。
据你所知，你的三叉戟落在一棵有 n 个结点以 1 号结点为根的有根树 T 。每个结点都有一个权值 a_i ，并且任意时刻所有结点的权值都恰好构成 $1 \sim n$ 的一个排列。对于树上的两个结点 u, v ，记它们之间的简单路径为 $P(u, v)$ ，注意此处 $u, v \in P(u, v)$ 。
定义一个树上结点集合 S 为一个三叉，当且仅当存在三个结点 F, x, y （可以相同），满足：

- F 是 x 的祖先或 $F = x$;
- F 是 y 的祖先或 $F = y$;
- $S = P(F, x) \cup P(F, y)$;

你需要按顺序处理 q 次操作，操作共有三种：

操作 1：引雷，给定两个结点 x 和 y ，交换它们当前的权值。

操作 2：审判，给定两个不同的权值 $l < r$ ，翻转 $[l, r]$ 区间内的权值。

形式化地，对所有的权值进行如下的变化，设操作前结点 x 的权值为 a_x ，操作后为 a'_x ，定义

$$a'_x = \begin{cases} l + r - a_x, & l \leq a_x \leq r, \\ a_x, & \text{otherwise.} \end{cases}$$

随后对所有 $1 \leq x \leq n$ 的整数 x ，同时令 $a_x \leftarrow a'_x$ ，可以证明操作前后序列始终是 $1 \sim n$ 的一个排列。

操作 3：寻戟，给定 $l \leq r$ ，判断 $S = \{x \in \mathbb{N}, 1 \leq x \leq n \mid l \leq a_x \leq r\}$ 是否是一个三叉。

请依次回答所有操作 3：寻戟所提出的问题。

输入格式

第一行输入一个整数 T ($1 \leq T \leq 10^3$)，表示测试数据的数量。

每组测试数据第一行输入两个整数 n 和 q ($1 \leq n, q \leq 10^5$)，表示树上的结点数量和操作的数量。

随后 $n - 1$ 行每行输入两个整数 u 和 v ($1 \leq u, v \leq n, u \neq v$)，表示树上存在一条连接 u 和 v 的边。

随后一行输入 n 个整数 $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq n$)，表示每个结点的权值。

随后 q 行每行表示一次操作，第 i 行首先输入一个整数 op_i ($1 \leq op_i \leq 3$)，表示操作类型，随后格式如下：

- 若 $op_i = 1$ ，随后输入两个整数 x 和 y ($1 \leq x, y \leq n$)，表示进行一次操作 1：引雷，交换 x 和 y 的权值。
- 若 $op_i = 2$ ，随后输入两个整数 l 和 r ($1 \leq l < r \leq n$)，表示进行一次操作 2：审判，翻转权值在 $[l, r]$ 的点的权值。
- 若 $op_i = 3$ ，随后输入两个整数 l 和 r ($1 \leq l \leq r \leq n$)，表示进行一次操作 3：寻戟，查询权值在 $[l, r]$ 之间的点是否构成一个三叉。

保证单组测试数据内，所有结点的权值构成 $1 \sim n$ 的一个排列，给定的边形成一棵 n 个结点的树。

保证所有测试数据的 $\sum n \leq 3 \times 10^5$ ， $\sum q \leq 3 \times 10^5$ 。

输出格式

对于每组测试数据中指令类型为操作 **3**：寻戟的指令输出一行一个字符串：

- 若 $S = \{x \in \mathbb{N}, 1 \leq x \leq n \mid l \leq a_x \leq r\}$ 是一个三叉，则输出 YES；
- 反之则输出 NO。

注意评测时不会区分 YES 和 NO 的大小写，换言之当答案是肯定的时候输出 yes、YES、Yes、YeS 等都会被认为是正确的。

样例 1

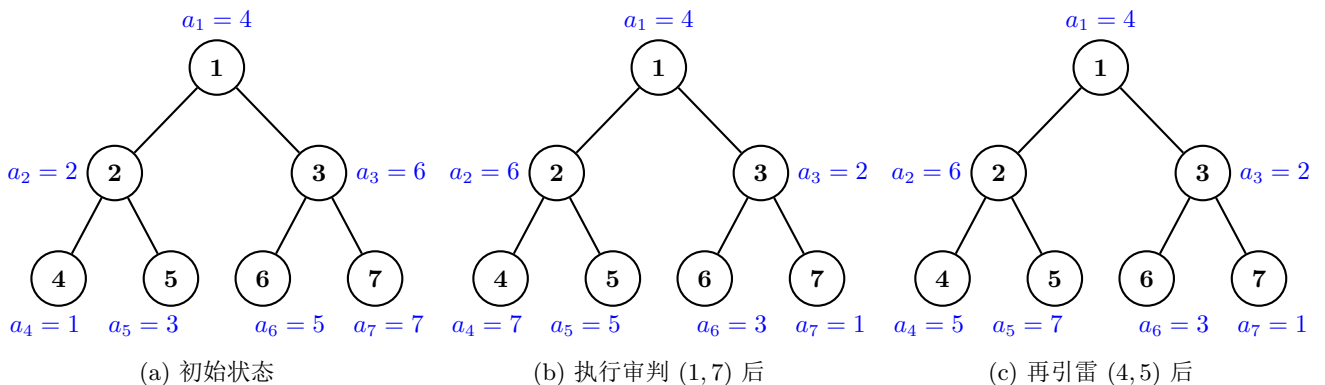
standard input

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

standard output

```
YES
NO
YES
YES
```

样例 1 解释



对于第 1 组测试数据的第三次寻戟，经过前一次审判操作后，当前各结点的权值为

$$(a_1, a_2, \dots, a_7) = (4, 6, 2, 7, 5, 3, 1).$$

此时权值 4, 5, 6, 7 分别位于结点 1, 5, 2, 4，因此 $S = \{1, 2, 4, 5\}$ 。

取 $F = 1$ ， $x = 4$ ， $y = 5$ ，依次验证条件：

- 结点 1 是结点 4 的祖先。
- 结点 1 是结点 5 的祖先。
- $P(1, 4) \cup P(1, 5) = \{1, 2, 4\} \cup \{1, 2, 5\} = S$ 。

因此 S 是一个三叉。