必可赛前公益众筹赛

第四试

时间：2024 年 10 月 5 日 07:40 ~ 12:10

题目名称	平衡的白鸽	你也在这里	卜卦	都是假的
题目类型	传统型	传统型	传统型	交互型
目录	dove	meet	divining	fake
可执行文件名	dove	meet	divining	fake
输入文件名	dove.in	meet.in	divining.in	fake.in
输出文件名	dove.out	meet.out	divining.out	fake.out
每个测试点时限	4.0 秒	5.0 秒	6.0 秒	6.0 秒
内存限制	512 MiB	1024 MiB	1024 MiB	512 MiB
测试点数目	50	20	12	6
测试点是否等分	是	是	否	否

提交源程序文件名

对于 C++ 语言	dove.cpp	meet.cpp	divining.cpp	fake.cpp

编译选项

对于 C++ 语言	-O2 -std=c++14

注意事项

1. 选手请直接提交源程序至 becoder.com.cn 上的对应比赛。
2. 输入输出文件名必须使用英文小写。
3. C++ 中函数 main() 的返回值类型必须是 int，值必须为 0。
4. 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
5. 程序可使用的栈空间大小与该题内存空间限制一致。
6. 若无特殊说明，每道题的代码大小限制为 100KB。
7. 若无特殊说明，输入与输出中同一行的相邻整数、字符串等均使用一个空格分隔。
8. 输入文件中可能存在行末空格，请选手使用更完善的读入方式（例如 scanf 函数）避免出错。
9. 直接复制 PDF 题面中的跨页样例，数据将带有行序号和页眉页脚，建议选手直接使用对应目录下的样例文件进行测试。
10. 使用 std::deque 等 STL 容器时，请注意其内存空间消耗。
11. 若发现有原题，请**保持静默状态直至比赛结束**，否则作违反考试纪律处理，**禁赛 3 年**。

平衡的白鸽（dove）

【题目背景】

看机械的白鸽 从空中飞过
要如何点睛 它才堪称鲜活
数字的晨昏 是否更缤纷
仿生的情人 是否更忠贞
推开一扇门 还有万千重门

——《蜃楼》

【题目描述】

小范有 n 只机械白鸽和 $2n$ 个翅膀，翅膀的重量为 w_i。经过研究，他得出了两个参数 A, B：

- 若白鸽两个翅膀的重量差大于 A，它将自行湮灭；
- 若白鸽两个翅膀的重量差大于 B 且小于等于 A，它是一只普通的白鸽；
- 若白鸽两个翅膀的重量差小于等于 B，它是一只鲜活的白鸽。

现在，小范已经制成了所有的翅膀，并将它们的重量尽数告知你，希望你能合理地分配，使得不存在湮灭的白鸽，且鲜活的白鸽尽量多。若无法做到，输出 -1。

在不同的平行世界，你需要解决不同的询问。

【输入格式】

从文件 ***dove.in*** 中读入数据。

为了减少读入量，请注意此题特殊的输入格式。

第一行两个正整数 T, n，分别表示数据组数和该测试点中机械白鸽的数量。

第二行 $2n$ 个正整数 w_i，表示翅膀的重量。

每组数据的 w 均由上组经过一些修改得到。

对于每组数据：

第一行一个整数 p，表示修改的数量。

接下来 p 行，每行两个正整数 x, v，表示将 w_x 修改为 v。

最后一行两个正整数 A, B，表示题中的参数。

询问不是独立的。

【输出格式】

输出到文件 ***dove.out*** 中。

对于每组数据，如果存在合法方案，输出一个正整数，表示“鲜活的白鸽”的最大数量。否则，输出 -1。

【样例 1 输入】

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

【样例 1 输出】

```
2
1
```

【样例 1 解释】

初始，有 $w = \{1, 2, 3, 4\}$。

对于第一组数据，不存在修改，$w = \{1, 2, 3, 4\}, A = 3, B = 1$。其中 $(1, 2)$ 配对，$(3, 4)$ 配对。

对于第二组数据，$w_1 \leftarrow 6$，$w_2 \leftarrow 1$，$w = \{6, 1, 3, 4\}, A = 5, B = 1$，其中 $(6, 1)$ 配对，$(3, 4)$ 配对。

【样例 2】

见选手目录下的 ***dove/dove2.in*** 与 ***dove/dove2.ans***。

该样例符合测试点 $1 \sim 5$ 的限制。

【样例 3】

见选手目录下的 ***dove/dove3.in*** 与 ***dove/dove3.ans***。

该样例符合测试点 $21 \sim 30$ 的限制。

【样例 4】

见选手目录下的 ***dove/dove4.in*** 与 ***dove/dove4.ans***。

该样例符合测试点 31 ∼ 40 的限制。

【子任务】

对于 100% 的数据，$1 \le n \le 5 \times 10^5$，$1 \le T \le 20$，$1 \le x \le 2n$，$0 \le p \le 2 \times 10^4$，$1 \le w_i, v \le 10^{18}$，$1 \le B \le A \le 10^{18}$。

每个测试点的具体限制见下表：

测试点	$T \le$	$n \le$	$w_i \le$	特殊性质
$1 \sim 5$	5	5	10^{18}	否
$6 \sim 10$	1	10		否
$11 \sim 15$	20	5×10^5		是
$16 \sim 20$			10^5	否
$21 \sim 30$		$2,000$	10^{18}	否
$31 \sim 40$		10^5		否
$41 \sim 50$		5×10^5		否

特殊性质：$A = B$。

你也在这里（meet）

【题目背景】

啊 那一个人 是不是只存在梦境里
为什么我用尽全身力气 却换来半生回忆

若不是你渴望眼睛 若不是我救赎心情
在千山万水人海相遇 喔 原来你也在这里
——刘若英《原来你也在这里》

【题目描述】

如何在梦里找寻他的踪迹？小烟有一个可以任意变化大小的正方形梦境搜索仪，他希望利用这个科技实现一场相遇。

小烟的梦境是一个 n 行 m 列的方格平面，仪器可以在这个平面上沿水平或竖直方向移动。仪器移动过程中覆盖过的地方，即可被认为是**搜索过的**。

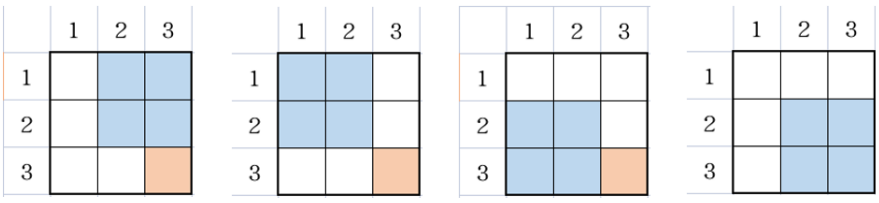
为了保证梦境的完整性，小烟选择了 k 个方格作为**禁行区**，仪器移动过程中，不能覆盖到这 k 个方格。

小烟可以任意选择一个 1 到 10^{100} 内的正整数 x，让这个仪器变形为一个占地 x 行 x 列的正方体。然后他会选择一个位置放置这个仪器，并按照他喜欢的方式操纵这个仪器**沿水平或竖直方向**移动。

小烟认为，仪器的尺寸越大，搜索的效率就越高。因此，请你帮他找到一个最大的正整数 x，使得存在一种操纵仪器的方式，满足下列两个条件：

- 仪器移动的过程中，它不会覆盖到禁行区内任意一个方格的任意一个部分；
- 仪器移动有限时间后，所有非禁行区的方格均是被**搜索过的**。

下面四张图展示了一个 3 行 3 列的梦境，其中 $(3,3)$ 是唯一的一个禁行区（用橙色表示）。有一个占地 2 行 2 列的仪器放置在梦境中（用浅蓝色表示）。



图一 图二 图三 图四

- 仪器可以从图一的位置向左移动至图二的位置，再向下移动至图三的位置。在移动结束后，所有非禁行区的方格均是被**搜索过的**。

- 仪器不能从图一的位置向下移动至图四的位置，因为这样移动会导致仪器覆盖 $(3,3)$ 这个禁行区。

由于梦境总是在变化，所以你需要处理多组数据。

【输入格式】

从文件 ***meet.in*** 中读入数据。

本题的单个测试点中有多组测试数据。

第一行两个正整数 id, T，表示测试点编号和数据组数。

接下来是 T 组测试数据。

每组的第一行三个正整数 n, m, k，表示梦境的尺寸以及禁行区的数量。

接下来的 k 行每行两个正整数 a_i, b_i，表示 (a_i, b_i) 代表的方格是一个禁行区。

【输出格式】

输出到文件 ***meet.out*** 中。

对于每组数据，输出一行一个整数，表示最大的、满足条件的正整数 x。若不存在满足条件的正整数，请输出 `-1`。

【样例 1 输入】

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

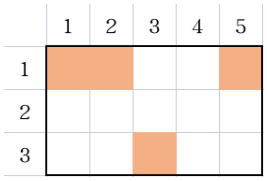
```
5 6
6 5
6 6
```

【样例 1 输出】

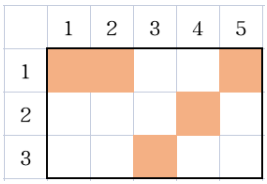
```
2
1
-1
3
```

【样例 1 解释】

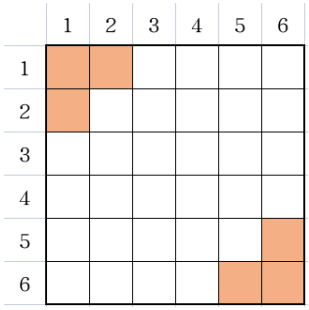
- 第一组数据描述的梦境和题目描述部分的图例一致。$x = 2$ 时，可以按照题目描述部分描述的方式移动。
- 第二组数据描述的梦境如下图所示。



- 第三组数据描述的梦境如下图所示。$(3, 1)$ 和 $(3, 5)$ 所在的两个连通块互相不能到达，故不存在一个合法的正整数 x。



- 第四组数据描述的梦境如下图所示。



【样例 2】

见选手目录下的 ***meet/meet2.in*** 与 ***meet/meet2.ans***。

【样例 2 解释】

该样例满足测试点 $9, 10$ 的限制。

【样例 3】

见选手目录下的 ***meet/meet3.in*** 与 ***meet/meet3.ans***。

【样例 3 解释】

该样例满足测试点 $1 \sim 3$ 的限制。

【样例 4】

见选手目录下的 ***meet/meet4.in*** 与 ***meet/meet4.ans***。

【样例 4 解释】

该样例满足测试点 8 的限制。

【样例 5】

见选手目录下的 ***meet/meet5.in*** 与 ***meet/meet5.ans***。

【样例 5 解释】

该样例满足测试点 12 的限制。

【样例 6】

见选手目录下的 ***meet/meet6.in*** 与 ***meet/meet6.ans***。

【样例 6 解释】

该样例满足测试点 $18 \sim 20$ 的限制。

【子任务】

下面设 $S=\sum(n\times m)$。

对于 100% 的数据，保证 $1\le n,m\le 3\times 10^6$，$0\le k<n\times m\le 3\times 10^6$，$1\le T\le S\le 10^7$，$\sum k\le 3\times 10^6$。

测试点编号	S	特殊性质
$1\sim 3$	≤ 500	/
$4\sim 6$	$\le 3,000$	
7	$\le 2\times 10^5$	A
8		B
$9,10$		/
11	$\le 10^6$	A
12		C
$13,14$		/
$15,16$	$\le 10^7$	B
17		C
$18\sim 20$		/

特殊性质 A：保证 $k\le 1$。

特殊性质 B：保证 $n\le 3$。

特殊性质 C：保证在同一个测试点中，a_i 全部相等。

【提示】

```
1  该隐瞒的事总清晰 千言万语只能无语
2  爱是天时地利的迷信 喔 原来你也在这里
```

请注意常数因子对程序效率的影响。

卜卦（divining）

【题目背景】

那年仲夏 你背上行囊离开家
古道旁 我欲语泪先下
田里庄稼 收获了一茬又一茬
而 我们何时发芽

不停的猜 猜 猜 又卜了一卦
吉凶祸福 还是担惊受怕
对你的爱 爱 爱 望断了天涯
造化弄人 缘分阴错阳差
——崔子格《卜卦》

【题目描述】

“合卦”是一种双人卦，由一个标有 $1\sim n$ 的盘面和编号为 $1\sim n$ 的 n 根卦签组成。卜卦时，两人会按顺序依次执行下列操作：

1. 将 n 根卦签均匀打乱并随机分成两堆。假设两堆中的卦签根数分别为 x 和 $n-x$，其中 $x\in[0,n]$。
2. 第一个人选择卦签数量为 x 的一堆，并将这一堆中的所有卦签按任意顺序放置在盘面的 $1\sim x$ 号位置上。
3. 第二个人将剩的那一堆中的所有卦签按任意顺序放置在盘面的 $x+1\sim n$ 号位置上。
4. 卦象可以用一个数 B 来表示。具体地，B 是盘面的所有位置上，卦签编号与位置编号的差的绝对值之和。形式化地，设盘面上位置 i 处放置的卦签编号为 p_i，有 $B=\sum_{i=1}^{n}|i-p_i|$。

小烟和小嘉正在卜卦。小烟先将签数为 x 的那一堆卦签摆在了盘面上 $1\sim x$ 的位置上，其中第 i 个位置放置的卦签编号为 p_i。

通过阅读古书，小嘉认为卦象为 b 是最好的。小嘉希望知道，是否存在一种放置剩下的 $n-x$ 根卦签的方式，使得最终的卦象为 b。

小烟有一个幸运数字 s。下面的问题中将会用到这个数字。

本题有两类测试点。

- 在第一类测试点中，$sid=1$。此时你只需要**判断是否存在**至少 s 种放置剩下的 $n-x$ 根卦签的方式，使得最终的卦象为 b 即可。
- 在第二类测试点中，$sid=2$。你需要判断是否存在至少 s 种放置剩下的 $n-x$ 根卦签的方式，使得最终的卦象为 b。若存在，你需要找到放置结束后，满足卦象

为 b 的所有盘面 p 中，**字典序第 s 小**的盘面。

由于小烟和小嘉占卜了很多次，所以你需要处理多组不同的数据。

【输入格式】

从文件 ***divining.in*** 中读入数据。

本题的单个测试点中有多组测试数据。

第一行三个整数 id, sid, T，表示测试点编号、测试点种类和数据组数。

接下来是 T 组测试数据。

每组的第一行四个非负整数 n, b, s, x，表示盘面大小、目标卦象、小烟的幸运数字和小烟放置的卦签数量。

第二行 x 个正整数，第 i 个正整数 p_i 表示小烟在盘面上位置 i 处放置的卦签编号。

【输出格式】

输出到文件 ***divining.out*** 中。

对于每组数据，按照下面的要求进行输出：

- $sid = 1$ 时：输出一行一个整数。若存在，则输出 `1`；否则输出 `-1`。
- $sid = 2$ 时：若存在，则输出两行，第一行一个整数 `1`，第二行一个长度为 n 的排列 q，表示字典序第 s 小的合法最终盘面。若不存在，输出一行一个整数 `-1` 即可。

【样例 1 输入】

```
12 2 5
3 2 3 0
3 2 1 1
2
3 2 1 2
2 3
3 4 2 0
3 4 2 1
3
```

【样例 1 输出】

```
-1
1
```

```
2 1 3
-1
1
3 1 2
1
3 2 1
```

【样例 1 解释】

当 $n = 3$ 时，所有可能的最终盘面有以下的、按字典序从小到大排序的 6 种：

- $(1, 2, 3)$，$B = 0$。
- $(1, 3, 2)$，$B = 2$。
- $(2, 1, 3)$，$B = 2$。
- $(2, 3, 1)$，$B = 4$。
- $(3, 1, 2)$，$B = 4$。
- $(3, 2, 1)$，$B = 4$。

下面给出每组数据的解释：

- 对第一组数据，$B = 2$ 的最终盘面只有 2 个，不足 3 个。
- 对第二组数据，满足 $p_1 = 2$ 且 $B = 2$ 的最终盘面共有 1 个，字典序第 1 小的为 $(2, 1, 3)$。
- 对第三组数据，没有满足 $p_1 = 2$，$p_2 = 3$ 且 $B = 2$ 的最终盘面。
- 对第四组数据，满足 $B = 4$ 的最终盘面共有 3 个，字典序第 2 小的为 $(3, 1, 2)$。
- 对第五组数据，满足 $p_1 = 3$ 且 $B = 4$ 的最终盘面共有 2 个，字典序第 2 小的为 $(3, 2, 1)$。

【样例 2】

见选手目录下的 ***divining/divining2.in*** 与 ***divining/divining2.ans***。

【样例 2 解释】

该样例满足测试点 $12, 13$ 的限制。

【样例 3】

见选手目录下的 ***divining/divining3.in*** 与 ***divining/divining3.ans***。

【样例 3 解释】

该样例满足测试点 4 的限制。

【样例 4】

见选手目录下的 ***divining/divining4.in*** 与 ***divining/divining4.ans***。

【样例 4 解释】

该样例满足测试点 $9 \sim 11$ 的限制。

【样例 5】

见选手目录下的 ***divining/divining5.in*** 与 ***divining/divining5.ans***。

【样例 5 解释】

该样例满足测试点 $14, 15$ 的限制。

【样例 6】

见选手目录下的 ***divining/divining6.in*** 与 ***divining/divining6.ans***。

【样例 6 解释】

该样例满足测试点 $16, 17$ 的限制。

【样例 7】

见选手目录下的 ***divining/divining7.in*** 与 ***divining/divining7.ans***。

【样例 7 解释】

该样例满足测试点 $20 \sim 23$ 的限制。

【样例 8】

见选手目录下的 ***divining/divining8.in*** 与 ***divining/divining8.ans***。

【样例 8 解释】

该样例满足测试点 $24 \sim 27$ 的限制。

【子任务】

下面设 $S=\sum n$。

对于 100% 的数据，保证 $1\le n\le 28$，$1\le S\le 100$，$0\le b\le n^2$，$1\le s\le 10^{18}$，$0\le x\le n$。保证 $1\le p_i\le n$ 且 p_i 互不相同。

本题的各个测试点不等分。

测试点	sid	n	S	特殊性质	单个测试点分值	总分值
1	$=1$	≤ 9	$\le 10^2$	/	3	3
2, 3		≤ 14			4	8
4		≤ 28		A, B		4
5, 6				A		8
7, 8				B	7	14
$9\sim 11$				/	5	15
12, 13	$=2$	≤ 9			4	8
14, 15		≤ 14				
16, 17		≤ 17			2	4
18, 19		≤ 22				
$20\sim 23$		≤ 25			3	12
$24\sim 27$		≤ 28				

特殊性质 A：保证 $s=1$。

特殊性质 B：保证 $x=0$ 。

【提示】

```
猜 猜 猜 又卜了一卦
是上上签 可还是放不下
对你的爱 爱 挨过几个冬夏
日夜思念 祈求别再变卦
```

请注意常数因子对程序效率的影响。

都是假的（fake）

这是一道交互题。

【题目背景】

图 1: “六年的感情比不过比赛的规则”

【题目描述】

Simons 有一个 n 行 m 列的矩阵 a，其中每个元素都是 $[1, n \times m]$ 的整数。矩阵的下标从 1 开始，坐标 (x, y) 表示矩阵第 x 行第 y 列的元素。

若将矩阵 a 分割成若干个 r 行 c 列的子矩阵（分割需满足 $r|n, c|m$，每个元素恰好属于一个子矩阵），这些矩阵全部相等，则 Simons 称数对 (r, c) 不是假的。

你需要猜出 Simons 手中的矩阵 a 有多少对 (r, c) 不是假的。

你只能询问 Simons 原矩阵两个不相交的子矩阵是否完全相同。两个子矩阵不相交，当且仅当它们没有覆盖到位置相同的元素。

你最多只能询问 $3 \times \lfloor \log_2(n + m) \rfloor$ 次。

因为 Simons 的朋友们也想玩矩阵，而他们的感情不是假的，所以你需要解决多组数据。

【交互方式】

你不需要也不应该直接读入或输出任何数据。你不需要也不应该实现主函数。

交互开始前，交互库会得到测试数据组数 T。

接着进行 T 次交互。每次交互库会从输入文件中读取 n, m 和矩阵 a。这也说明了交互库不是自适应的：矩阵 a 在交互开始前就已确定。

你需要实现一个函数 `int truth(int n, int m, int lim)`，n, m 为矩阵的大小，lim 为该测试数据中你的询问次数上限，返回值是你得出的答案。

在每个测试点中，交互库都会调用恰好 T 次 `truth` 函数解决每组测试数据。**因此请注意你的程序清空问题。**

你的实现可以调用函数 `int simons(int h, int w, int px1, int py1, int px2, int py2)`。函数会返回 a 矩阵的以坐标 $(px1, py1)$ 和坐标 $(px2, py2)$ 为左上角的两个 h 行 w 列子矩阵是否完全相等（是则返回 1，不是则返回 0）。特别地，请注意你的调用需要满足：

- 不能超出矩阵的边界
- 两个子矩阵不能相交
- 调用不能超过限制次数
- $h > 0 \wedge w > 0$

违背这些条件的调用会使函数返回 -1。交互库将在发生这样的调用后对该测试数据判定为错误，但需要注意返回 -1 后继续运行程序仍然可能导致意外的后果。

每次调用 `truth` 函数时，你最多只能调用 lim 次 `simons` 函数。

【测试细节】

提交时你的程序应当附带 `fake.h` 头文件。

题目保证在各函数规定的调用次数限制下，交互库运行的时间不超过 1.5 秒，交互库使用的内存大小不超过 20MiB。本地测试时请注意交互库对你的程序效率表现产生的影响。

我们在下发文件中下发了交互库 `grader.cpp`，本题需要的头文件 `fake.h`，编译程序 `compile_cpp.sh/.bat` 和示例代码 `sample.cpp`。你可以在示例代码的基础上编写程序。

下面介绍本地测试方法。

对于 Windows 系统：

- 将下发文件中的程序全部移动到你的工作文件夹,并将你的代码命名为 `fake.cpp`，然后运行 `compile_cpp.bat`。
- 若通过编译，此时你的工作文件夹下会出现 `fake.exe`。运行后向通过**标准输入**输入样例输入，正常情况下，交互库会向屏幕上打印你的程序交互后返回的答案，若出现 -1 意味着你的程序进行了不合法调用。将此输出与样例输出比较即可。

对于 Linux 系统：

- 将你的栈空间手动调至 512MiB。
- 将下发文件中的程序全部移动到你的工作文件夹,并将你的代码命名为 `fake.cpp`，然后运行 `compile_cpp.sh`。
- 若通过编译，此时你的工作文件夹下会出现可执行文件 `fake`。运行后向通过**标准输入**输入样例输入，正常情况下，交互库会向屏幕上打印你的程序交互后返回

的答案，若出现 -1 意味着你的程序进行了不合法调用。将此输出与样例输出比较即可。

你也可以通过修改 `grader.cpp` 的方式改为文件输入输出。

注意：

- 修改下发文件对最终测试是**无效操作**。最终测试只会收取你编写的 `fake.cpp`。
- 你不应通过非法方式获取交互库的内部信息，如试图直接读取矩阵 a 的值，或直接与标准输入、输出流进行交互。**此类行为将被视为作弊**。
- 最终测试的交互库与下发交互库有所不同，**你的解法不应该依赖交互库实现**。

【输入格式】

你不需要直接从输入文件中读入任何数据。这里描述的是交互库读入的数据格式。

第一行一个整数 T，表示数据组数。

对于每组数据，第一行三个整数 n, m, lim，表示矩阵大小和 `simons` 函数调用限制。

接下来 n 行 m 列描述矩阵 a。

【输出格式】

你不需要直接向输出文件中输出任何数据。这里描述的是交互库输出的数据格式。

T 行整数，表示你的程序得出的答案。特别地，如果你的程序发生不合法调用，交互库将输出 -1。

【样例 1 输入】

```
2
3 4 360
1 2 1 2
3 3 3 3
2 1 2 1
1 3 90
1 1 1
```

【样例 1 输出】

```
2
2
```

【样例 1 解释】

本题的输入输出样例为对交互库的样例输入与其对应的期望输出。

对于第一组数据的矩阵 a，$(3,4)$ 和 $(3,2)$ 不是假的，共 2 对。

对于第二组数据的矩阵 a，$(1,1)$ 和 $(1,3)$ 不是假的，共 2 对。下面是 `truth` 函数中一个可能的能确定答案且合法的交互流程：

次数	调用	返回值
1	simons(1,1,1,1,1,2)	1
2	simons(1,1,1,1,1,3)	1

【样例 2】

见选手目录下的 ***fake/fake2.in*** 与 ***fake/fake2.ans***。

这组样例满足了子任务 2 的数据范围。

【样例 3】

见选手目录下的 ***fake/fake3.in*** 与 ***fake/fake3.ans***。

这组样例满足了子任务 3 的数据范围。

【样例 4】

见选手目录下的 ***fake/fake4.in*** 与 ***fake/fake4.ans***。

这组样例满足了子任务 6 的数据范围。

【部分分设置】

本题开启子任务捆绑和子任务依赖。

对于 100% 的数据，保证 $1 \le n, m \le 1000, 1 \le a_{ij} \le n \times m$，$1 \le T \le 10$。$lim$ 的限制较为特殊，如下表所示。

子任务	n	m	特殊性质	lim	得分	子任务依赖
1	≤ 30	≤ 30	无特殊限制	$= 30nm$	10	无
2	≤ 1	$\le 10^3$	无特殊限制	$= 3\times \lfloor \log_2(n+m) \rfloor$	20	无
3	$\le 10^3$		A		5	无
4			B		5	无
5	≤ 729	≤ 729	C		20	无
6	$\le 10^3$	$\le 10^3$	无特殊限制		40	1,2,3,4,5

特殊性质 A：a 中所有元素相同。

特殊性质 B：所有 a_{ij} 都独立地在数据范围内等概率随机。

特殊性质 C：n, m 均为 3 的正整数幂。