

[📑 题目列表 \(/contest/438/home\)](/contest/438/home)[📊 提交状态 \(/contest/438/submissions\)](/contest/438/submissions)[🏆 排行榜 \(/contest/438/ranklist/normal\)](/contest/438/ranklist/normal)[❓ 提问 \(/contest/438/issue\)](/contest/438/issue)

理发店

时间限制: 1.0 秒

刷新 ↺

空间限制: 512 MiB

题目描述

理发店同时来了 n 名顾客，你作为理发店的店长，需要为每名顾客先洗发再剪发。已知服务第 i 名顾客洗发需要 a_i 分钟，剪发需要 b_i 分钟。

你需要遵循如下限制：

- 由于人手有限，同时只有最多一位顾客在洗发；同样地，同时只有最多一位顾客在剪发；
- 对于某一位顾客，必须在洗发结束后才能开始剪发；

另外，由于这些顾客都是同时来的，你可以按照任意顺序为他们服务，没有先来后到的区别。请求出完成所有服务的最少时间。

输入格式

从标准输入读入数据。

一个输入文件中会包含多个输入数据。输入文件的第一行为一个正整数 T ，表示输入数据的组数。

每个输入数据的第一行包含一个整数 n ，表示顾客的数量。接下来 n 行，每行包含两个正整数 a_i 和 b_i ，表示第 i 位顾客洗发和剪发所需的时间。

输出格式

输出到标准输出。

输出 T 行，每行包含一个整数，表示你对该输入数据求出的最少时间。

样例1输入

```
2
3
10 20
25 40
20 30
3
30 35
25 25
25 30
```

样例1输出

```
100
115
```

样例1解释

对于第一个输入数据，一种可能的最优方案如下：

- 0 分钟到 10 分钟，为第 1 位顾客洗发；
- 10 分钟到 30 分钟，为第 3 位顾客洗发；
- 10 分钟到 30 分钟，为第 1 位顾客剪发；
- 31 分钟到 56 分钟，为第 2 位顾客洗发；
- 30 分钟到 60 分钟，为第 3 位顾客剪发；
- 60 分钟到 100 分钟，为第 2 位顾客剪发。

对于第二个输入数据，一种可能的最优方案如下：

- 0 分钟到 25 分钟，为第 2 位顾客洗发；
- 25 分钟到 50 分钟，为第 2 位顾客剪发；
- 25 分钟到 50 分钟，为第 3 位顾客洗发；
- 50 分钟到 80 分钟，为第 3 位顾客剪发；
- 50 分钟到 80 分钟，为第 1 位顾客洗发；
- 80 分钟到 115 分钟，为第 1 位顾客剪发。

子任务

本题共有 10 个测试点，每个测试点 10 分。对于所有输入数据，保证 $1 \leq T \leq 100$ ，每个数据的 $0 \leq n \leq 5$ ，且 $1 \leq a_i, b_i \leq 100$ 。

- 测试点 1 满足 $n = 0$ ；
- 测试点 2 满足 $n = 1$ ；
- 测试点 3、4 满足 $n = 2$ ；
- 测试点 5、6 满足 $n = 3$ ；
- 测试点 7、8 满足 $n = 4$ ；
- 测试点 9、10 满足 $n = 5$ 。

语言和编译选项

#	名称	编译器	额外参数	代码长度限制
0	g++	g++	-O2 -std=c++14 -DONLINE_JUDGE	65536 B
1	gcc	gcc	-O2 -std=c11 -DONLINE_JUDGE	65536 B
2	java	javac		65536 B
3	python2	python		65536 B
4	python3	python3		65536 B

递交历史		
#	状态	时间
191975	Accepted	2023-09-04 16:23:31
191922	Wrong Answer	2023-09-04 16:16:00
191884	Wrong Answer	2023-09-04 16:08:04
191860	Wrong Answer	2023-09-04 16:04:09
191832	Wrong Answer	2023-09-04 15:58:28
191830	Wrong Answer	2023-09-04 15:57:47
191776	Compilation Error	2023-09-04 15:43:24
191743	Wrong Answer	2023-09-04 15:32:43
191725	Wrong Answer	2023-09-04 15:28:43
191632	Wrong Answer	2023-09-04 15:09:53
		12

飞镖

时间限制：3.0 秒

空间限制：512 MiB

刷新 

题目描述

小 R 在一个 $n \times m$ 的网格中玩飞镖游戏，其中的每个格子可以用二维坐标 (x, y) 指定（其中 $1 \leq x \leq n, 1 \leq y \leq m$ ）， x 轴以向下为递增正方向，而 y 轴以向右为递增正方向。网格中的每个格子上都可以有至多一个**标记**，初始时所有格子均没有**标记**。

小 R 会依次投掷 q 枚飞镖，每一枚飞镖由三个参数 x, y, t 指定，其中 (x, y) 表示其投掷位置， t 为飞镖类型，为 $+$ \times $*$ 三者之一，具体说明如下：

对于所有类型的飞镖，都会在投掷时向 (x, y) 位置写入一个对应**标记**（如 $+$ 型飞镖会在 (x, y) 位置写入 $+$ ）。如果此位置原本已有**标记**，则会将其覆盖。

对于 $+$ 类型飞镖，投掷时会向投掷位置的上、下、左、右的四个方向各发射一只小飞镖。这些小飞镖会沿着各自的方向以每秒 1 格的速度一直飞行，直到**其中之一**碰撞到了某个已有的**标记**或者超出网格边界。当碰撞或出界发生时，所有的小飞镖会同时停止飞行。对于发生碰撞的小飞镖，会将其所在位置的**标记**清除。对于未发生碰撞也未出界的小飞镖，会将其所在位置**标记**为 $+$ 。

对于 \times 类型飞镖，投掷时会向投掷位置的左上、左下、右上、右下的四个方向各发射一只小飞镖，速度为每秒 $\sqrt{2}$ 格。其余规则同上，但对于未发生碰撞也未出界的小飞镖，会将其所在位置**标记**为 \times 。

对于 $*$ 类型飞镖，投掷时会向投掷位置的上、下、左、右、左上、左下、右上、右下的八个方向各发射一只小飞镖（上下左右方向的速度为每秒 1 格，而斜线方向的速度为每秒 $\sqrt{2}$ 格。其余规则同上，但对于未发生碰撞也未出界的小飞镖，会将其所在位置**标记**为 $*$ 。

下面将举例说明：

例如 $n = 5, m = 6$ ，初始时棋盘为空，我们用 $.$ 表示一个空位置：

```
.....
.....
.....
.....
.....
```

假设向位置 $(2, 2)$ 投掷一枚 $+$ 类型的飞镖，则会先在 $(2, 2)$ 位置写入 $+$ ：

```
.....
.+.....
.....
.....
.....
```

然后，会向上、下、左、右四个方向各发射一只小飞镖，它们会以每秒 1 格的速度飞行。向上、向左的两只小飞镖会在 2 秒后出界，而此时向下、向右的两只小飞镖会停止飞行并将其所在位置**标记**为 +：

```
.....
.++.
.....
.+....
.....
```

假设在此基础上，向位置 (3, 3) 投掷一枚 x 类型的飞镖，则会先在 (3, 3) 位置写入 x：

```
.....
.++.
..x...
.+....
.....
```

然后，会向左上、左下、右上、右下四个方向各发射一只小飞镖，它们会以每秒 $\sqrt{2}$ 格的速度飞行。向左上、左下、右上的三只小飞镖会在 1 秒后碰撞到已有的标记，而此时向右下的一只小飞镖会停止飞行并将其所在位置**标记**为 x：

```
.....
.....
..x...
...x..
.....
```

假设在此基础上，向位置 (5, 4) 投掷一枚 * 类型的飞镖，则会先在 (3, 3) 位置写入 *：

```
.....
.....
..x...
...x..
...*..
```

然后，会向上、下、左、右、左上、左下、右上、右下八个方向各发射一只小飞镖，它们会以每秒 1 格或 $\sqrt{2}$ 格的速度飞行。向上的小飞镖会在 1 秒后碰撞到已有的标记，与此同时向下、左下、右下的三只小飞镖会出界，其余方向的小飞镖会停止移动，最终棋盘状态如下：

```
.....
.....
..x...
.*.*.
..***.
```

假设在此基础上，向位置 (3, 3) 投掷一枚 x 类型的飞镖，则最终状态如下（注意覆盖了该位置的原有标记）：

```
x...x.
.....
..x...
.*.*.
x.**..
```

本题目将会给出 n, m, q 和 q 个飞镖的信息，你需要对于投掷的每个飞镖回答：

- 该飞镖是否覆盖了投掷位置 (x, y) 的原有标记？如果覆盖了，原有的标记是什么？
- 该飞镖产生的小飞镖飞行了多少秒的时间？

- 产生的每个小飞镖的最终状态是什么？（碰撞到了标记/出界/未碰撞也未出界）

输入格式

从标准输入读入数据。

第一行输入三个整数 n, m, q ，以空格分隔。

接下来 q 行，每行输入三个元素 x, y, t ，以空格分隔，表示一枚飞镖的信息。

输出格式

输出到标准输出。

输出 q 行，每行输出三个元素，以空格分隔，表示一枚飞镖的信息：

- 如果该飞镖覆盖了投掷位置 (x, y) 的原有标记，则输出原有的标记字符，否则输出 `.`；
- 输出该飞镖产生的小飞镖飞行了多少秒的时间；
- 输出产生的每个小飞镖的最终状态，为 4 个或 8 个字符，`+` 型飞镖按照上、下、左、右的顺序输出，`x` 型飞镖按照左上、左下、右上、右下的顺序输出，`*` 型飞镖按照上、下、左、右、左上、左下、右上、右下的顺序输出，如果该小飞镖碰撞到了标记则输出标记字符，如果该小飞镖出界则输出 `o`，如果该小飞镖未碰撞也未出界则输出 `.`。

样例1输入

```
5 6 4
2 2 +
3 3 x
5 4 *
3 3 x
```

样例1输出

```
. 2 o.o.
. 1 +++
. 1 xo...o.o
x 2 ...*
```

子任务

本题共 25 个测试点，每个测试点 4 分。对于所有的数据， $1 \leq n, m, q \leq 10^5$ ，飞镖的位置都是均匀随机生成的。

- 测试点 1~3: $n, m, q \leq 1000$ ，所有的飞镖均为 `+` 类型；
- 测试点 4~6: $n, m, q \leq 1000$ ，所有的飞镖均为 `x` 类型；
- 测试点 7~9: $n, m, q \leq 1000$ ，所有的飞镖均为 `*` 类型；
- 测试点 10~11: $n, m, q \leq 1000$ ；
- 测试点 12~15: 所有的飞镖均为 `+` 类型；
- 测试点 16~19: 所有的飞镖均为 `x` 类型；
- 测试点 20~23: 所有的飞镖均为 `*` 类型；
- 测试点 24~25: 无特殊条件。

语言和编译选项				
#	名称	编译器	额外参数	代码长度限制
0	g++	g++	-O2 -std=c++14 -DONLINE_JUDGE	65536 B
1	gcc	gcc	-O2 -std=c11 -DONLINE_JUDGE	65536 B
2	java	javac		65536 B
3	python2	python		65536 B
4	python3	python3		65536 B

递交历史			
#	状态	时间	
192552	Runtime Error	2023-09-04 17:25:28	
192535	Wrong Answer	2023-09-04 17:23:58	
192475	Wrong Answer	2023-09-04 17:16:52	
192433	Wrong Answer	2023-09-04 17:10:58	
192407	Wrong Answer	2023-09-04 17:08:32	
192331	Wrong Answer	2023-09-04 17:01:04	
192277	Wrong Answer	2023-09-04 16:56:33	
192259	Wrong Answer	2023-09-04 16:54:59	
			1

[📑 题目列表 \(/contest/438/home\)](/contest/438/home)[📊 提交状态 \(/contest/438/submissions\)](/contest/438/submissions)[🏆 排行榜 \(/contest/438/ranklist/normal\)](/contest/438/ranklist/normal)[❓ 提问 \(/contest/438/issue\)](/contest/438/issue)

最小值网络

时间限制: 1.0 秒

刷新 ↺

空间限制: 512 MiB

题目描述

喜欢胡思乱想的小 I 最近学习了排序网络，于是发明了一个类似的网络：最小值网络。在最小值网络中，你需要通过若干个比较器将最小值放在最底下。

具体地， n 个变量的最小值网络是由若干个比较操作 $\text{Compare}(a_x, a_y) (1 \leq x < y \leq n)$ 组成的程序，这个程序满足，无论 n 个变量 a_1, a_2, \dots, a_n 的取值如何，按照如下的程序运行方式，程序运行完成后 a_1 对应所有变量中的最小值：

- 依次考虑网络中的每个比较操作。对于每个比较操作 $\text{Compare}(a_x, a_y)$ ，若 $a_x > a_y$ ，交换 a_x 与 a_y 。

然而小 I 注意到，他手中的比较器有一定概率会失效，所以小 I 希望当网络中某一个比较器失效的时候网络还是一个最小值网络。具体地，称一个最小值网络是**可靠的**当且仅当对于任意一个网络中的比较操作，删除这个操作保留其他操作及它们的顺序，得到的网络仍然是个最小值网络。

小 I 现在已经搭好了一些比较器，也就是决定好了一些比较操作。小 I 不希望修改之前搭的部分，因此你只能在小 I 决定的比较操作之后加入若干操作。小 I 想知道至少要加多少个比较操作才能让网络变成可靠的。

输入格式

从标准输入读入数据。

输入的第一行两个整数 n, m ，表示变量的数量和小 I 决定好的比较操作数量。

接下来 m 行每行两个整数 x, y ，表示一个比较操作 $\text{Compare}(a_x, a_y)$ 。

输出格式

输出到标准输出。

输出一行一个整数表示最少需要在已经决定的比较操作后添加多少个比较操作才能让这个网络变为可靠的。

样例1输入

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


样例1输出

3

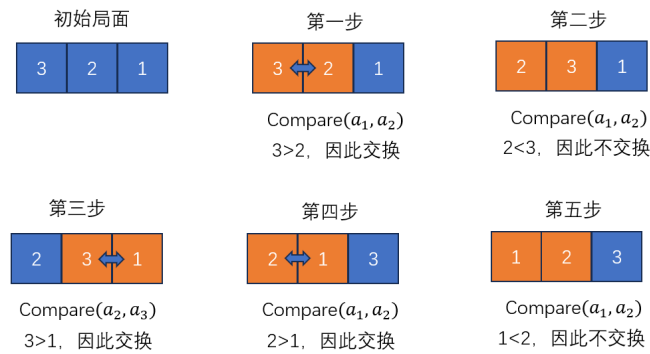
样例1解释

在已有的网络后加入 $\text{Compare}(a_2, a_3)$, $\text{Compare}(a_1, a_2)$ 和 $\text{Compare}(a_1, a_2)$ 三个操作, 可以证明得到的网络是可靠的。

考虑一个例子: 若第四个比较操作被删除, 那么剩余的网络为

$\text{Compare}(a_1, a_2), \text{Compare}(a_1, a_2), \text{Compare}(a_2, a_3), \text{Compare}(a_1, a_2), \text{Compare}(a_1, a_2)$.

当 $(a_1, a_2, a_3) = (3, 2, 1)$ 时, 下图展示了网络的运行, 可以发现最终 $a_1 = 1$ 是最小值, 符合要求。



可以证明加入两个比较操作后网络不可能变为可靠的, 因此输出为 3。

子任务

对于所有测试数据, $1 \leq n \leq 1000, 0 \leq m \leq 4000, 1 \leq x, y \leq n$ 。

子任务编号	$n \leq$	$m \leq$	特殊性质	分值
1	4	8	×	20
2	50	200		
3	10^3	0		10
4		4,000	√	20
5			×	30

特殊性质: $x = 1$ 。

语言和编译选项

#	名称	编译器	额外参数	代码长度限制
0	g++	g++	-O2 -std=c++14 -DONLINE_JUDGE	65536 B
1	gcc	gcc	-O2 -std=c11 -DONLINE_JUDGE	65536 B

#	名称	编译器	额外参数	代码长度限制
2	java	javac		65536 B
3	python2	python		65536 B
4	python3	python3		65536 B

递交历史

#	状态	时间
193180	Wrong Answer	2023-09-04 17:59:45
193142	Wrong Answer	2023-09-04 17:58:36
193047	Wrong Answer	2023-09-04 17:55:06
193040	Wrong Answer	2023-09-04 17:54:50
192995	Wrong Answer	2023-09-04 17:53:09
192980	Compilation Error	2023-09-04 17:52:37
192951	Wrong Answer	2023-09-04 17:51:49
192915	Wrong Answer	2023-09-04 17:49:50
192795	Wrong Answer	2023-09-04 17:43:22