



# 2012 亚太地区信息学奥林匹克竞赛

## APIO 2012

**竞赛时间：2012年5月12日8:15-13:15**

题目名称	派遣	守卫	苦无
目录	dispatching	guard	kunai
每个测试点时限	1 秒	1 秒	3 秒
内存限制	256 MB	256 MB	256 MB
总分	100	100	100
输入文件名	标准输入（键盘）		
输出文件名	标准输出（屏幕）		

提交源程序须加后缀

对于 C++ 语言	dispatching.cpp	guard.cpp	kunai.cpp
对于 C 语言	dispatching.c	guard.c	kunai.c
对于 Pascal 语言	dispatching.pas	guard.pas	kunai.pas

编译器版本及编译选项

语言	国际评测版本	国内评测版本	编译选项
C++	g++ version 4.6.3	g++ version 4.5.5	-O2 -lm
C	gcc version 4.6.3	gcc version 4.5.5	-O2 -lm
Pascal	fpc version 2.4.4	fpc version 2.4.2	-O2 -Sd -Sh

## 派遣

### 【问题描述】

在一个忍者的帮派里，一些忍者们被选中派遣给顾客，然后依据自己的工作获取报偿。

在这个帮派里，有一名忍者被称之为 Master。除了 Master 以外，每名忍者都有且仅有一个上级。为保密，同时增强忍者们的领导力，所有与他们工作相关的指令总是由上级发送给他的直接下属，而不允许通过其他方式发送。

现在你要招募一批忍者，并把它们派遣给顾客。你需要为每个被派遣的忍者支付一定的薪水，同时使得支付的薪水总额不超过你的预算。另外，为了发送指令，你需要选择一名忍者作为管理者，要求这个管理者可以向所有被派遣的忍者发送指令，在发送指令时，任何忍者（不管是否被派遣）都可以作为消息的传递人。管理者自己可以被派遣，也可以不被派遣。当然，如果管理者没有被派遣，你就不需要支付管理者的薪水。

你的目标是在预算内使顾客的满意度最大。这里定义顾客的满意度为派遣的忍者总数乘以管理者的领导力，其中每个忍者的领导力也是一定的。

写一个程序，给定每一个忍者  $i$  的上级  $B_i$ ，薪水  $C_i$ ，领导力  $L_i$ ，以及支付给忍者们的薪水总预算  $M$ ，输出在预算内满足上述要求时顾客满意度的最大值。

### 【数据范围】

$1 \leq N \leq 100,000$	忍者的个数；
$1 \leq M \leq 1,000,000,000$	薪水总预算；
$0 \leq B_i < i$	忍者的上级的编号；
$1 \leq C_i \leq M$	忍者的薪水；
$1 \leq L_i \leq 1,000,000,000$	忍者的领导力。

对于 30% 的数据， $N \leq 3000$ 。

### 【输入格式】

从标准输入读入数据。

第一行包含两个整数  $N$  和  $M$ ，其中  $N$  表示忍者的个数， $M$  表示薪水的总预算。

接下来  $N$  行描述忍者们的上级、薪水以及领导力。其中的第  $i$  行包含三个整数  $B_i, C_i, L_i$  分别表示第  $i$  个忍者的上级，薪水以及领导力。Master 满足  $B_i = 0$ ，并且每一个忍者的上级的编号一定小于自己的编号  $B_i < i$ 。

### 【输出格式】

输出到标准输出。

输出一个数，表示在预算内顾客的满意度的最大值。

**【样例输入】**

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

**【样例输出】**

```
6
```

**【样例说明】**

如果我们选择编号为 1 的忍者作为管理者并且派遣编号为 3 和编号为 4 的忍者，薪水总和为 4，没有超过总预算 4。因为派遣了 2 个忍者并且管理者的领导力为 3，用户的满意度为  $2 \times 3 = 6$ ，是可以得到的用户满意度的最大值。

## 守卫

### 【问题描述】

APIO 王国正被忍者攻击！忍者非常厉害，因为他们在进攻的时候可以躲在阴影里面使得其他人看不到他们。整个王国除了国王居住的 APIO 城堡以外都已经被占领了。在城堡前，有  $N$  个灌木丛，从 1 到  $N$  编号，有  $K$  个忍者躲在恰好  $K$  个灌木丛后面。APIO 城堡里有  $M$  个守卫。守卫  $i$  监视着编号从  $A_i$  到  $B_i$  的连续的一段灌木丛。每个守卫都向国王报告在他所监视范围内是否有忍者出现。作为国王的仆人，你需要告诉国王，基于守卫的报告，哪些灌木丛后面一定躲着一个忍者，即对于任何和守卫报告不矛盾的忍者排列方式，在这个灌木丛后面都躲着一个忍者。

你需要写一个程序来输出所有的这些灌木丛的编号。

### 【数据范围】

$1 \leq N \leq 100,000$	灌木的数量；
$1 \leq K \leq N$	忍者数；
$1 \leq M \leq 100,000$	守卫数。

对于 10% 的数据， $N \leq 20$ ， $M \leq 100$ ；

对于 50% 的数据， $N \leq 1000$ ， $M \leq 1000$ 。

### 【输入格式】

从标准输入读入数据。

第一行包含三个用空格分隔的整数  $N$ ,  $K$ ,  $M$ ， $N$  是灌木丛的个数， $K$  是忍者的个数， $M$  是守卫的个数。

接下来  $M$  行，每行描述一个守卫的信息。其中的第  $i$  行包含三个整数  $A_i$ ,  $B_i$ ,  $C_i$ ，表示第  $i$  个守卫的监视范围是从  $A_i$  到  $B_i$  ( $A_i \leq B_i$ )。  $C_i$  是 0 或者 1，若是 0 表示范围内没有看到忍者，1 表示范围内有至少一个忍者。

输入数据保证至少存在一种忍者排列方式满足所有条件。

### 【输出格式】

输出到标准输出。

若存在灌木丛，在其后面一定躲着忍者，则将这些一定躲着忍者的灌木丛按照编号从小到大的顺序依次输出，每个一行。即若有  $X$  个这样的灌木丛，则需要输出  $X$  行。若不存在，则输出一行一个“-1”，不包含引号。

**【样例输入 1】**

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

**【样例输出 1】**

```
3
5
```

**【样例说明 1】**

在这个样例中，有两种可能的安排方式：1, 3, 5 或者 2, 3, 5。即 3 和 5 后面必然躲着一个忍者。

考虑第一个灌木丛，存在一种安排方案使得它的后面躲着忍者，但也存在一种安排方案使得它后面没有躲忍者，因此不应该输出 1。同理，不应该输出 2。

**【样例输入 2】**

```
5 1 1
1 5 1
```

**【样例输出 2】**

```
-1
```

**【样例说明 2】**

在这个样例中，没有灌木丛后面一定躲着忍者。

# 苦无

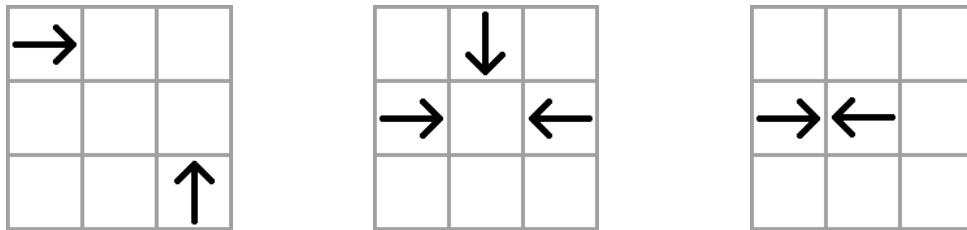
## 【问题描述】

苦无(Kunai)是一种忍者使用的形状像刀的武器，忍者通过投掷苦无攻击对手。

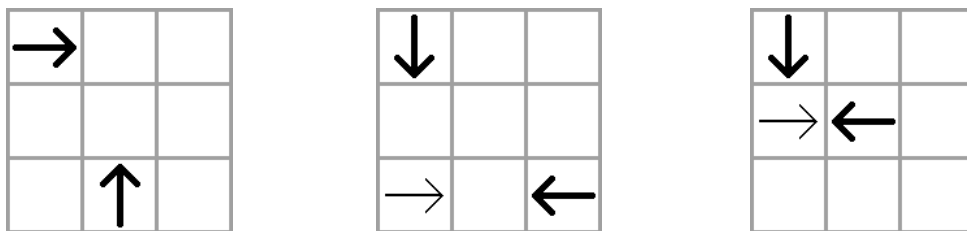
现在有  $N$  名忍者聚集在一块  $H$  行  $W$  列的棋盘式的广场上。每个忍者都站在其所在方块的中心处，任何两个忍者都不在同一个方块上。每个忍者都拿着一个苦无，面朝上、下、左、右四个方向中的一个方向站着。在时刻 0，所有忍者同时向其所朝向的方向投掷苦无。

每个苦无将会一直保持其初始的方向，并以单位速度飞行。如果某个时刻一个位置上多于一个的苦无，它们将会相撞并且消失。苦无特别小，可以看成质点。同时，由于忍者的移动速度特别快，他们不会被苦无击中。

在下面的例子中，我们用箭头来表示苦无，而箭头的方向即为苦无的方向。在这些图中，所有的苦无都会相撞后消失。



在下面的图中，两个粗线箭头表示的苦无不会相撞。其中在第二个和第三个图中，其中一个粗线表示的苦无会与细线表示的苦无相撞后消失，因此不会撞上另一个粗线表示的苦无。



你的任务是计算经过足够长的时间之后，在这个  $W \times H$  的广场中有多少格子被苦无经过。

## 【数据范围】

- |  |       |
|--|-------|
| $1 \leq N \leq 100,000$                | 忍者数；  |
| $1 \leq W \leq 1,000,000,000$          | 列数；   |
| $1 \leq H \leq 1,000,000,000$          | 行数；   |
| $1 \leq X_i \leq W, 1 \leq Y_i \leq H$ | 坐标范围。 |

在 10% 的数据中， $N \leq 1000, W \leq 1000, H \leq 1000$ 。

在 40% 的数据中， $N \leq 1000$ 。

### 【输入格式】

从标准输入读入数据。

第一行包含两个被空格隔开的整数  $W, H$ ，表示广场的尺寸为  $W$  列  $H$  行。

第二行包含一个整数  $N$ ，表示忍者的数量。

接下来  $N$  行中，第  $i$  行有三个以空格分隔的整数  $X_i, Y_i, D_i$ ，表示第  $i$  个忍者处在从左往右的  $X_i$  列、从上往下的第  $Y_i$  行，任何两个忍者不在同一个位置。第  $i$  个忍者面向的方向由  $D_i$  表示，分别为：

- $D_i = 0$ ，表示忍者向右；
- $D_i = 1$ ，表示忍者向上；
- $D_i = 2$ ，表示忍者向左；
- $D_i = 3$ ，表示忍者向下。

### 【输出格式】

输出到标准输出。

输出一个整数，表示经过足够长的时间之后，在这个  $W \times H$  的广场中被苦无经过的格子数量。

### 【样例输入 1】

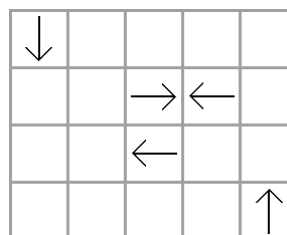
```
5 4
5
3 3 2
3 2 0
4 2 2
5 4 1
1 1 3
```

### 【样例输出 1】

```
11
```

### 【样例说明】

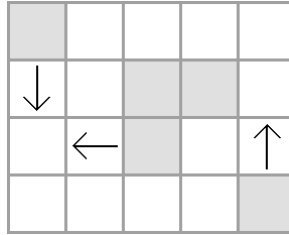
在时刻 0，苦无的情况如下图所示



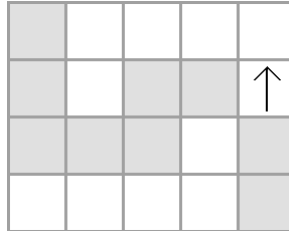
在下面的描述中，忍者  $i$  投掷的苦无将用苦无  $i$  表示。

在时刻 0.5，苦无 2 和苦无 3 相撞后消失。

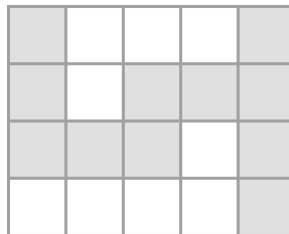
下图为时刻 1 的情况，加深的格子表示已经被苦无经过。



在时刻 2，苦无 1 和苦无 5 相撞后消失，此时的广场如下图所示。



之后没有苦无相撞。再经过足够时间后的广场如下图所示。



共有 11 个格子被苦无经过，因此输出 11。

**【样例输入 2】**

```

7 6
12
3 2 3
6 3 2
7 1 3
1 5 0
3 6 1
6 6 1
4 5 2
1 3 0
6 5 2
5 1 2
6 4 3
4 1 3

```

**【样例输出 2】**

29