

## 猫和老鼠 (game)

### 【题目描述】

《猫和老鼠》是一部家喻户晓的动画片。小 G 根据这部动画片设计了一个游戏。在这个游戏中，玩家需要帮助 Tom 使用机器猫抓住 Jerry。

Jerry 的活动范围是数轴上的区间  $[0, m]$ 。初始时刻（即第 0 秒时），Jerry 可能位于该区间内的任意位置。之后，它可以在该区间内自由移动，但任意时刻的速度不会超过每秒 1 单位长度。

Tom 有  $n$  个可供部署的机器猫，其中部署第  $i$  ( $1 \leq i \leq n$ ) 个机器猫的成本为  $w_i$ 。若部署了第  $i$  ( $1 \leq i \leq n$ ) 个机器猫，则其会在第  $t_i$  秒从位置  $a_i$  出现，然后以每秒 1 单位长度的速度向位置  $b_i$  匀速移动，并在抵达后消失。

Jerry 初始拥有  $k$  点生命值。每当它与一个机器猫完全重合时（即存在某个时刻，两者位置完全相同），其生命值将减少 1，该机器猫也会随之失效。当 Jerry 的生命值小于或等于 0 时，Tom 即成功抓住 Jerry。

小 G 设定 Tom 必须在初始时刻就部署好机器猫。因此，玩家需要在游戏开始前选定若干个机器猫进行部署。玩家获胜当且仅当部署好机器猫后，对于 Jerry 所有可能的移动路径，Tom 均能成功抓住 Jerry。

小 G 为这个游戏设计了很多关卡，并邀请你进行测试。为了控制游戏难度，小 G 计划为部署机器猫的成本总和设置一个合理的上限。你需要帮助小 G 求出，玩家获胜所需部署的机器猫的成本总和的最小值。

### 【实现细节】

选手不需要，也不应该实现 `main` 函数。

选手需要确保提交的程序包含头文件 `game.h`，即在程序开头加入以下代码：

```
1 #include "game.h"
```

选手需要在提交的程序源文件 `game.cpp` 中实现以下两个函数：

```
1 void init(int c, int t);
```

- $c, t$  分别表示测试点编号与测试数据组数。 $c = 0$  表示该测试点为样例。
- 对于每个测试点，该函数会在程序开始运行时被交互库调用恰好一次。

```
1 long long game(int n, int m, int k, std::vector<int> a,  
                std::vector<int> b, std::vector<int> t, std::vector<int> w);
```

- $n, m, k$  分别表示机器猫的个数、Jerry 的活动范围与 Jerry 的初始生命值。
- 对于  $0 \leq i < n$ ， $a_i, b_i, t_i, w_i$  分别表示第  $i + 1$  个机器猫的出现位置、最终位置、出现时间与部署成本。

- 该函数需要返回成本总和的最小值。特别地，如果部署所有机器猫也无法获胜，则返回  $-1$ 。
- 对于每个测试点，该函数会被交互库调用恰好  $t$  次。

注意：在任何情况下，交互库运行所需时间均不会超过 0.1 秒，所用内存为固定大小，且均不超过 64 MiB。

### 【测试程序方式】

试题目录下的 `grader.cpp` 是交互库参考实现，最终测试时所用的交互库实现与该参考实现有所不同，因此选手的解法不应该依赖交互库实现。

选手可以在本题目目录下使用如下命令编译得到可执行程序：

```
1 g++ grader.cpp game.cpp -o game -O2 -std=c++14 -static
```

对于编译得到的可执行程序：

- 可执行文件将从标准输入读入以下格式的数据：
  - 输入的第一行包含两个非负整数  $c, t$ ，分别表示测试点编号和测试数据组数。
  - 接下来依次为每组测试数据，对于每组测试数据：
    - \* 第一行包含三个正整数  $n, m, k$ ，分别表示机器猫的个数、Jerry 的活动范围与 Jerry 的初始生命值。
    - \* 第  $i + 1$  ( $1 \leq i \leq n$ ) 行包含四个非负整数  $a_i, b_i, t_i, w_i$ ，分别表示第  $i$  个机器猫的出现位置、最终位置、出现时间与部署成本。
- 可执行文件将输出以下格式的数据至标准输出：
  - 对于每组测试数据，输出一行一个整数，表示成本总和的最小值。特别地，如果部署所有机器猫也无法获胜，则输出  $-1$ 。

### 【样例 1 输入】

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

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

### 【样例 1 输出】

```
1 6
2 -1
3 35
```

### 【样例 1 解释】

该样例共包含三组测试数据。

对于第一组测试数据, 玩家可以选择部署第 1, 2, 3 个机器猫, 成本总和为  $1+2+3=6$ 。若玩家选择部署第 1, 3 个机器猫, 则当 Jerry 初始时刻位于位置 7, 且从第 6 秒开始以每秒 1 单位长度的速度匀速移动至位置 1 时, Tom 无法成功抓住 Jerry。

对于第二组测试数据, 当 Jerry 初始时刻位于位置 5.5 且不移动时, 它只会与第 1 个机器猫在第 3.5 秒时完全重合, 因此部署所有机器猫也无法获胜。

对于第三组测试数据, 玩家可以选择部署第 1, 2, 3, 4, 5, 7 个机器猫, 成本总和为  $7+8+9+3+3+5=35$ 。

### 【样例 2】

见选手目录下的 *game/game2.in* 与 *game/game2.ans*。

该样例满足测试点 3, 4 的约束条件。

### 【样例 3】

见选手目录下的 *game/game3.in* 与 *game/game3.ans*。

该样例满足测试点 5, 6 的约束条件。

**【样例 4】**

见选手目录下的 `game/game4.in` 与 `game/game4.ans`。  
该样例满足测试点 10 ~ 12 的约束条件。

**【样例 5】**

见选手目录下的 `game/game5.in` 与 `game/game5.ans`。  
该样例满足测试点 16 ~ 18 的约束条件。

**【样例 6】**

见选手目录下的 `game/game6.in` 与 `game/game6.ans`。  
该样例满足测试点 23, 24 的约束条件。

**【样例 7】**

见选手目录下的 `game/game7.in` 与 `game/game7.ans`。  
该样例满足测试点 25 的约束条件。

**【下发文件说明】**

在本试题目录下：

1. `grader.cpp` 是提供的交互库参考实现。
2. `game.h` 是头文件，选手不用关心具体内容。
3. `template_game.cpp` 是提供的示例代码，选手可参考并实现自己的代码。

选手注意对所有下发文件做好备份。最终评测时只测试本试题目录下的 `game.cpp`，对该程序以外文件的修改不会影响评测结果。

**【数据范围】**

设  $N, S$  分别为单个测试点内所有测试数据的  $n, nk$  的和。对于所有测试数据，均有：

- $1 \leq t \leq 20$ ;
- $1 \leq n \leq 5 \times 10^4, N \leq 3 \times 10^5$ ;
- $1 \leq m \leq 10^9, 1 \leq k \leq 10, S \leq 10^6$ ;
- 对于所有  $1 \leq i \leq n$ ，均有  $0 \leq a_i, b_i \leq m, 0 \leq t_i, w_i \leq 10^9$ 。

测试点编号	$n \leq$	$k \leq$	特殊性质
1, 2	10	10	A
3, 4	$10^3$	1	BC
5, 6	$5 \times 10^4$		CD
7, 8	$10^3$	10	C
9	$5 \times 10^4$		CD
10 ~ 12			C
13, 14	$10^3$	1	无
15	$5 \times 10^4$		D
16 ~ 18			10
19 ~ 22	80		
23, 24	300		
25	$5 \times 10^4$		

特殊性质 A:  $m \leq 10$ , 且对于所有  $1 \leq i \leq n$ , 均有  $t_i, w_i \leq 10$ 。

特殊性质 B:  $m \leq 10^3$ , 且对于所有  $1 \leq i \leq n$ , 均有  $t_i \leq 10^6$ 。

特殊性质 C: 对于所有  $1 \leq i \leq n$ , 均有  $w_i = 0$ 。

特殊性质 D: 对于所有  $1 \leq i \leq n$ , 均有  $a_i \leq b_i$ 。

### 【评分方式】

注意:

- 选手不应当通过非法方式获取交互库的内部信息, 如直接与标准输入、输出流进行交互。此类行为将被视为作弊;
- 最终的评测交互库与样例交互库的实现不同。

本题首先会受到和传统题相同的限制, 例如编译错误会导致整道题目得 0 分, 运行时错误、超过时间限制、超过空间限制等会导致相应测试点得 0 分等。选手只能在程序中访问自己定义的变量以及交互库给出的变量, 尝试访问其他地址空间将可能导致编译错误或运行错误。

在上述条件基础上:

- 对于每个测试点, 程序得到满分当且仅当每次调用 `game` 函数时返回的答案均正确。