

# NOIP 模拟赛

(请选手务必仔细阅读本页内容)

## 一、题目概况

中文题目名称	团队竞技	甲虫	两种货币	长途巴士
英文题目名称	battle	beetle	currencies	bus
可执行文件名	battle	beetle	currencies	bus
输入文件名	battle.in	beetle.in	currencies.in	bus.in
输出文件名	battle.out	beetle.out	currencies.out	bus.out
提交文件名	battle.cpp	beetle.cpp	currencies.cpp	bus.cpp
每个测试点时限	2秒	0.2秒	4秒	2秒
测试点数目	20	20	20	20
每个测试点分值	5	5	5	5
内存限制	1024MB	256MB	1024MB	256MB
题目类型	传统题	传统题	传统题	传统题

## 二、编译命令

题目名称	battle	beetle	currencies	bus
对于 C++ 语言	-o battle battle.cpp -lm -std=c++14 -O2 -Wl,--stack=2147483647	-o beetle beetle.cpp -lm -std=c++14 -O2 -Wl,--stack=2147483647	-o currencies currencies.cpp -lm -std=c++14 -O2 -Wl,--stack=2147483647	-o bus bus.cpp -lm -std=c++14 -O2 -Wl,--stack=2147483647

## 三、注意事项

- 文件夹名、文件名(程序名和输入输出文件名)必须使用英文小写。
- C/C++中函数main()的返回值类型必须是int,程序正常结束时的返回值必须是0。
- 统一评测时采用的机器配置为：windows下lemon评测。
- 请尽力优化，会收获更多的部分得分。

## 团队竞技(battle)

## 题目描述

JOI 大学有  $N$  只海狸，他们都参与竞技编程。每只海狸有三项能力值：思考值，行动值和运气值。如果一个能力值很大，意味着他这项能力比较强大。对于第  $i$  ( $i \in [1, N]$ ) 只海狸，他的思考值为  $X_i$ ，行动值为  $Y_i$ ，运气值为  $Z_i$ 。

今年 JOI 大学的海狸们将参与一场团体竞技编程，一支队伍由三名队员组成。Bitaro 是 JOI 大学的教练，由于团队合作很重要，Bitaro 决定从  $N$  只海狸中选出三只海狸组成队伍，这三只海狸要满足以下条件：

**条件：**每个成员都有自己的优势，这意味着每个成员都有一项能力值严格大于其他两人的对应能力值。

在所有符合条件的组队中，Bitaro 想要选一个总能力最强的队伍，一个队伍的总能力定义为：三人最大思考值，三人最大行动值和三人最大运气值之和。

请你求出，是否存在一个符合条件的组队，如果是，计算队伍总能力可能的最大值。

## 输入格式

第一行一个整数  $N$  表示海狸数。

接下来  $N$  行每行三个整数  $X_i, Y_i, Z_i$  表示海狸的各项能力值。

## 输出格式

一行一个整数，如果不存在符合条件的组队，输出 `-1`，否则输出队伍总能力的最大值。

## 样例

### 样例1输入

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

### 样例1输出

```
13
```

### 样例1解释

由海狸 1, 4, 5 组成的队伍符合条件，因为：

1. 海狸 1 的优势是运气。
2. 海狸 4 的优势是行动。
3. 海狸 5 的优势是思考。

总能力值为： $5 + 4 + 4 = 13$ 。

可以证明这是符合条件的组队中，总能力值最高的队伍。

注意如果选择海狸 1, 3, 5，总能力值将达到 15，但是这会导致海狸 1 没有特长。

### 样例2输入

```
8
1 1 1
1 1 5
1 5 1
5 1 1
1 5 5
5 1 5
5 5 1
5 5 5
```

### 样例2输出

```
15
```

### 样例2解释

最优组队为：海狸 2, 3, 4。

### 样例3输入

```
4
1 2 3
1 2 3
1 2 3
1 2 3
```

### 样例3输出

```
-1
```

## 数据范围

对于所有数据，满足：

- $3 \leq N \leq 150000$ 。
- $1 \leq X_i, Y_i, Z_i \leq 10^8 \ (1 \leq i \leq N)$ 。

详细子任务附加限制及分值如下表所示：

子任务编号	附加限制
1 ~ 4	$N \leq 300$
5 ~ 10	$N \leq 4000$
11 ~ 12	$X_i, Y_i, Z_i \leq 5 \ (i \in [1, N])$
13 ~ 14	$X_i, Y_i, Z_i \leq 20 \ (i \in [1, N])$
15 ~ 16	$X_i, Y_i, Z_i \leq 300 \ (i \in [1, N])$
17 ~ 18	$X_i, Y_i, Z_i \leq 4000 \ (i \in [1, N])$
19 ~ 20	无附加限制

# 甲虫(beetle)

## 题目描述

请注意本题不同寻常的时间限制，建议使用高效的读入方式。

有  $N$  只甲虫在桌面上排成一行，从左到右编号为  $0 \sim N - 1$ 。第  $i$  只甲虫的**体力**为  $S_i$ 。

每次可以选择**连续的**至多  $K$  只甲虫，用  $E$  的力量去击打它们。被击打的甲虫，若体力**不大于**  $E$ ，则会死亡，否则无事发生。**死亡的甲虫会留在原地，可以击打死亡的甲虫。**

可以多次击打甲虫，每次  $E$  的大小可以不同。

定义  $f(i)$  为：杀死最左边的  $i$  只甲虫，需要力量**和**的最小值。对于  $i = 1, 2, \dots, N$ ，求出  $f(i)$ 。

为了减小输出量，你只需要求出  $\left(\sum_{1 \leq i \leq N} f(i) \cdot 23^{N-i}\right) \bmod (10^9 + 7)$ 。

## 输入格式

输入共两行。

输入第一行有两个数  $N, K$ 。

第二行包含  $N$  个数  $S_1, S_2 \dots S_N$ 。

## 输出格式

输出仅一个数，表示答案。

## 样例

### 样例1输入

```
8 3
3 2 9 8 7 11 3 4
```

## 样例1输出

720026253

## 样例1解释

$f$  的值分别为 3, 3, 9, 12, 12, 20, 23, 23。

## 数据范围

对于所有的数据： $1 \leq K \leq N \leq 2.5 \times 10^6$ ,  $1 \leq S_i \leq 2 \times 10^9$ 。

测试点编号	$N \leq$	特殊性质
1 ~ 5	$2 \times 10^3$	无
6 ~ 7	$5 \times 10^5$	无
8 ~ 10	$1 \times 10^6$	无
11 ~ 15	$2.5 \times 10^6$	保证 $S$ 在数据范围内随机
16 ~ 20	$2.5 \times 10^6$	无

# 两种货币(currencies)

## 题目描述

在 JOI 王国中，有  $n$  个城市，编号从 1 到  $n$ 。JOI 王国有  $n - 1$  条双向道路，编号从 1 到  $n - 1$ 。第  $i$  条道路连接城市  $a_i$  和城市  $b_i$ 。

在 JOI 王国中，一些道路上放有检查站。有  $m$  个检查站，编号从 1 到  $m$ 。第  $j$  个检查站位于道路  $p_j$  上。通过该检查站需要支付 1 枚金币或  $c_j$  枚银币。

在 JOI 王国有  $q$  名公民，编号从 1 到  $q$ 。第  $k$  名公民持有  $x_k$  枚金币和  $y_k$  枚银币，并希望从城市  $s_k$  前往城市  $t_k$ 。由于金币具有较高的价值，所有公民都希望尽可能多地保留金币。

编写一个程序，给定 JOI 王国中的城市、道路、检查站和公民信息，对于每个  $k(1 \leq k \leq q)$ ，判断公民  $k$  是否能够从城市  $s_k$  前往城市  $t_k$ ，并在此条件成立时计算公民  $k$  所能保留的最多金币数。

## 输入格式

第一行输入三个数  $n, m, q$ 。

接下来的  $n - 1$  行，每行输入两个数  $a_i, b_i$ 。

接下来的  $m$  行，每行两个数  $p_j, c_j$ 。

接下来的  $q$  行，每行四个数  $s_k, t_k, x_k, y_k$ 。

## 输出格式

输出  $q$  行，其中第  $k$  行中，如果公民  $k$  可以从城市  $s_k$  前往城市  $t_k$ ，请输出公民  $k$  可以保留的最多金币数。否则，在第  $k$  行中输出  $-1$ 。

## 样例

样例1输入

```
5 4 3
1 2
1 3
2 4
2 5
2 9
2 4
3 5
4 7
3 4 2 11
5 3 4 5
2 3 1 1
```

样例1输出

```
1
2
-1
```

样例2输入

```
10 7 9
1 8
6 3
5 9
7 9
3 1
3 4
10 1
2 6
5 6
9 4
7 4
7 4
2 4
7 4
7 4
1 4
8 6 5 3
3 9 8 0
4 7 6 15
7 4 9 3
6 4 8 0
9 10 5 16
5 3 2 4
2 8 4 3
6 1 3 3
```

样例2输出

```
3
6
6
7
7
3
1
2
2
```

数据范围

数据范围:  $2 \leq n \leq 10^5$ ,  $1 \leq m, q \leq 10^5$ ,  $1 \leq a_i, b_i \leq N$ ,  $1 \leq p_i \leq N - 1$ ,  $1 \leq c_j \leq 10^9$ ,  $1 \leq s_k, t_k \leq N$ ,  $s_k \neq t_k$ ,  $0 \leq x_k \leq 10^9$ ,  $0 \leq y_k \leq 10^{18}$ , 所有数都是整数, 所有城市连通。

测试点编号	特殊性质
1 ~ 4	$n, m, q \leq 2000$
5 ~ 8	$c_1 = c_2 = \cdots = c_m$
9 ~ 12	$a_i = i, b_i = i + 1$
13 ~ 20	无特殊性质

长途巴士 (bus)

题目描述

某长途巴士发车时刻为 0，到达终点的时刻为  $X$ 。车上装有饮水机，乘客和司机可以在车上装水喝。途中有  $N$  个服务站，依次编号为  $1 \dots N$ 。巴士到达服务站  $i (1 \leq i \leq N)$  的时间是  $S_i$ 。发车前，水箱是空的。在发车前你可以给饮水机加水，在服务站时也可以给饮水机加水，但是都要钱，水价为每升  $W$  元。假设水箱容量无限。

本次巴士有  $M$  名乘客（不含司机），乘客均在起点上车，不会中途下车。乘客  $j (1 \leq j \leq M)$  在时刻  $kT + D_j (k = 0, 1, 2, \dots)$  需要装 1 升水，在其他时刻不装水。保证  $1 \leq D_j < T$ 。司机在时刻  $kT (k = 0, 1, 2, \dots)$  需要装 1 升水，在其他时刻不装水。如果到终点之前，某一名乘客想装水时饮水机没水了，这名乘客会怒而下车，此时需要向这名乘客退  $C_j$  元。如果到终点之前，司机想装水时没水了，司机会怒而下车，这车就不开了。  
**保证不会出现两人在同一时刻需要装水的情况。保证在服务站或是到达终点时，不存在司机或乘客需要喝水。**

我们希望花销（买水的总费用与退的所有车费之和）尽可能小，并且把车开到终点。试求至少需要花销多少元。

输入格式

输入共  $N + M + 1$  行。  
第一行有五个整数  $X, N, M, W, T$ 。  
接下来的  $N$  行，每行一个整数，其中第  $i$  行的整数为  $S_i$ 。  
接下来的  $M$  行，每行一个两个整数，其中第  $j$  行的两个整数为  $D_j, C_j$ 。

# 输出格式

输出仅一行，表示答案。

## 样例

### 样例1输入

```
19 1 4 8 7
10
1 20
2 10
4 5
6 5
```

### 样例1输出

```
103
```

### 样例1解释

- 出发时装了 7 升水；
- 在时刻 0, 1, 2, 4, 6, 司机和乘客 1, 2, 3, 4 前后装水，饮水机还剩 2 升水；
- 在时刻 7, 8, 司机和乘客 1 先后装水，饮水机没水了；
- 在时刻 9, 乘客 2 需要装水，但是饮水机没水了，因此乘客 2 下车，需要向其退款；
- 在时刻 10, 车到了服务站，向饮水机加 4 升水，此时饮水机剩余 4 升水；
- 在时刻 11, 13, 14, 15, 乘客 3, 4, 司机和乘客 1 先后装水，饮水机没水了；
- 在时刻 18, 乘客 3 需要装水，但是饮水机没水了，因此乘客 3 下车，需要向其退款；
- 在时刻 19, 车到达终点，总计花销为  $8 \times (7 + 4) + (10 + 5) = 103$ ，可以证明这是最小花销。

### 样例2输入

```
105 3 5 9 10
59
68
71
4 71
6 32
7 29
3 62
2 35
```

### 样例2输出

```
547
```

### 样例3输入

```
1000000000000 1 1 1000000 6
999999259244
1 123456789
```



样例3输出

333333209997456789

数据范围

对于所有数据,  $1 \leq T \leq X \leq 10^{12}$ ,  $1 \leq N, M \leq 2 \times 10^5$ ,  $1 \leq W \leq 10^6$ ,  $1 \leq S_i < X$ ,  $1 \leq D_j < T$ ,  $1 \leq C_j \leq 10^9$ 。

测试点编号	$N, M \leq$
1 ~ 5	8
6 ~ 10	100
11 ~ 15	2000
16 ~ 20	$2 \times 10^5$