

务农政策(policy)

时间限制: 1 Sec 内存限制: 128 MB

题目描述

为了建设新建立的艾尔奇亚联邦，空和白决定弄出一块 $a \times b$ 的平整土地作为试验田种植新作物。然而史蒂芙表示，只有一块 $n \times m$ 大小的空闲土地，而且是不平整的，并不能满足空白的要求。愚蠢的「智商吉祥物」史蒂芙思考很久，才发现她可以找人去填平那些土地。

由于饥饿，每个人只能够把一块土地增加一个单位的高度，而史蒂芙又希望让最少的人参加劳作以减少人力的消耗，那么问题就来了，最少需要多少人才能整理出达到空白要求的土地呢？

空白看到史蒂芙这么辛苦的计算，不免又想着调戏一下史蒂芙，所以他们给出了 T 次询问，要求史蒂芙一一作答。

输入

第一行，包含两个整数 $n, m (1 \leq n, m \leq 1000)$ ，表示空间土地的行与列数；

接下来 n 行，每行包含 m 个整数 $H_{i,j} (1 \leq H_{i,j} \leq 10^9)$ ，表示该位置的
土地高度；

接下来一行，包含一个整数 $T (1 \leq T \leq 10)$ ，表示询问次数；

接下来 T 行，每行包含两个整数 $a, b (1 \leq a \leq n, 1 \leq b \leq m)$ ，表示
需要的平整土地的
行与列数。

输出

输出 T 行，每行包含一个整数，表示对于第 i 个询问，史蒂芙最少需要的人力数量。

样例输入

```
3 4
1 8 3 4
5 2 3 1
3 6 2 2
4
1 1
2 2
2 3
3 2
```

样例输出

```
0
4
15
9
```

提示

对于30%的数据， $1 \leq n, m \leq 50, T=5$ ；

对于70%的数据， $1 \leq n, m \leq 200, T=5$ ；

对于100%的数据， $1 \leq n, m \leq 1000, T=10, 1 \leq a \leq n, 1 \leq b \leq m$ 。

米仓 (ricehub)

Time Limit: 1S Memory Limit: 128MB

Description:

乡间有一条笔直而长的路称为“米道”。沿着这条米道上 R 块稻田，每块稻田的坐标均为一个 1 到 L 之间(含 1 和 L)的整数。这些稻田按照坐标以不减的顺序给出，即对于 $0 \leq i < R$ ，稻田 i 的坐标 $X[i]$ 满足 $1 \leq X[0] \leq \dots \leq X[R-1] \leq L$ 。

注意：可能有多块稻田位于同一个坐标上。

我们计划建造一个米仓用于储存尽可能多的稻米。和稻田一样，米仓将建在米道上，其坐标也是一个 1 到 L 之间的整数(含 1 和 L)。这个米仓可以建在满足上述条件的任一位置上，包括那些原来已有一个或多个稻田存在的位置。

在收获季节，每一块稻田刚好出产一满货车的稻米。为了将这些稻米运到米仓，需要雇用一位货车司机来运米。司机的收费是每一满货车运送一个单位的距离收取 1 元。换言之，将稻米从特定的稻田运到米仓的费用在数值上等于稻田坐标与米仓坐标之差的绝对值。不幸的是，今年预算有限，我们至多只能花费 B 元运费。你的任务是要帮我们找出一个建造米仓的位置，可以收集到尽可能多的稻米。

Input

第一行三个整数 R 、 L 、 B 。

接下来 R 行，从小到大排列，表示每个稻田的坐标。

注意：运送稻米的总预算可能会很高，预算的总数是以 64 位整数给出的，建议在运算中使用 64 位整数。在 C/C++ 中，请使用 `long long` 类型数据；在 Pascal 中，请用 `Int64` 类型数据。

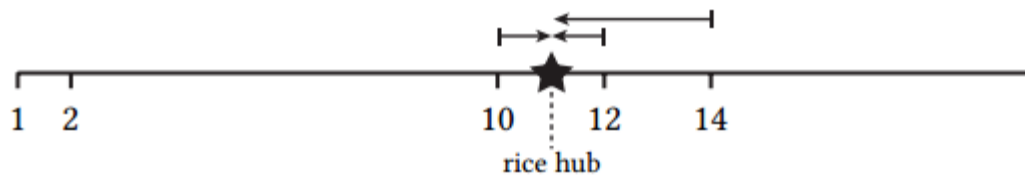
Output

输出一个整数表示最多能收集到的稻米。

Sample

5 20 6	3
1	
2	
10	
12	
14	

Hint



对于这个测试数据，米仓的最优放置位置有多种可能：你可以将它放置在 10 和 14 之间（含 10 和 14）的任何一个位置上(上图展示了多种最优放置中的一种)。这样，就可以将位于坐标 10,12 和 14 的稻田出产的米运到米仓。而对于其中任一最优的米仓位置而言，其总运输成本都不超过 6 元。很明显，在这个样例中，无论米仓选在何处，你最多只能收集到来 自 3 个稻田的稻米。

20%的数据， $1 \leq R \leq 100, 1 \leq L \leq 100, 0 \leq B \leq 10,000$

没有两个稻田在同一个坐标上。

40%的数据， $1 \leq R \leq 500, 1 \leq L \leq 10,000, 0 \leq B \leq 1,000,000$

70%的数据， $1 \leq R \leq 5,000, 1 \leq L \leq 1,000,000, 0 \leq B \leq 2,000,000,000$

100%的数据， $1 \leq R \leq 100,000, 1 \leq L \leq 1,000,000,000, 0 \leq B \leq 2,000,000,000,000,000$

圣杯战争(war)

时间限制: 2 Sec 内存限制: 256 MB

题目描述

“这场圣杯战争……从一开始，就很不对劲。”

由于这场圣杯战争过于异常，作为中立调停的「裁定者Ruler」贞德被大圣杯召唤。

圣杯战争在 n 座城市内进行，这些城市之间通过 m 条路径相互连接，由于某些特殊规则，所有的从者只能通过这 m 条路径从一个城市到达另一个城市。开始时，每个城市都会出现有且仅有一位从者。

令贞德不安的是，从者们既不是各自为战，也没有完整的阵营组合：每位从者分属一个阵营 k_i ，相同阵营的从者之间不会相互攻击，并且所有阵营均没有固定的从者数目。

在混战中，被击杀的从者会直接回归英灵座而消失，退出这场圣杯战争。大圣杯继续召唤一位英灵作为从者，让他/她降临在被击杀的从者所在的城市。

陷入迷惘的贞德整理了思绪，决定先去监督不同阵营的从者之间相距最近的一组战斗（指两个从者到达对方城市的最短距离），并希望知道每次更换（接着上一次的更新进行）了从者后所有战斗的最短距离。请帮助贞德完成这个任务。

大圣杯保证不会只剩一种阵营。

输入

第一行，包含四个整数 n, m, K, Q ，分别表示城市个数，路径条数，本次圣杯战争参战的不同阵营数目，以及更换从者的次数。

接下来 m 行，每行包含三个整数 u, v, w ，表示城市 u, v 之间存在一条长度为 w 的路径。

接下来一行，包含 n 个整数 k_1, k_2, \dots, k_n ，表示开始时每个城市召唤的从者所属阵营。

接下来 Q 行，每行包含两个整数 x, k ，表示城市 x 重新召唤了所属阵营为 k 的从者。

输出

对于每一次从者更换，输出一行包含一个整数，表示不同阵营之间战斗的最短距离。

样例输入

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

样例输出

```
1
3
3
1
```

提示

对于20%的数据， $1 \leq n, m \leq 1000, 1 \leq K \leq 10^3, 1 \leq Q \leq 100$

对于40%的数据， $1 \leq n \leq 5000, 1 \leq m \leq 10^4, 1 \leq K \leq 10^6, 1 \leq Q \leq 10^4$ 。

对于60%的数据， $1 \leq n \leq 2 \times 10^4, 1 \leq m \leq 4 \times 10^4, 1 \leq K \leq 10^6, 1 \leq Q \leq 10^4$ 。

对于100%的数据，
 $1 \leq u, v, x \leq n \leq 2 \times 10^5, 1 \leq m \leq 4 \times 10^5, 1 \leq k \leq K \leq 10^6, 1 \leq Q \leq 2 \times 10^5, 1 \leq w \leq 10^6$

最后四组数据中保证存在一组数据，每座城市最多只有10条路径连向其他城市。

当前的更改是在完成前面所有更改的前提下继续的。

数据保证n个城市之间，可以相互抵达