NOI 模拟赛 Day3

题目名称	重建: 地下铁道	走过安眠地的花丛	昔在、今在、永在的题目
题目类型	传统型	传统型	传统型
可执行文件名	`railway`	`death`	`problem`
输入文件名	`railway.in`	`death.in`	`problem.in`
输出文件名	`railway.out`	`death.out`	`problem.out`
时间限制	1.0 秒	1.0 秒	3.0 秒
内存限制	512 MiB	512 MiB	512 MiB

提交源程序文件名

对于 C++ 语言	`railway.cpp`	`death.cpp`	`problem.cpp`

编译选项

对于 C++ 语言	`-lm -O2 -std=c++14`

1. C++ 中函数 main() 的返回值类型必须是 int，值必须为 0。
2. 若无特殊说明，输入文件中同一行内的多个整数、浮点数、字符串等均使用一个空格进行分隔。
3. 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
4. 程序可使用的栈空间大小与该题内存空间限制一致。
5. 在终端下可使用命令 `ulimit -s unlimited` 将栈空间限制放大，但你使用的栈空间大小不应超过题目限制。
6. 对于因未遵守以上规则对成绩造成的影响，相关申诉不予受理。
7. 考试过程中若对题目有疑问，请联系出题人。
8. 题目很简单，AK 了请勿大声喧哗。

重建: 地下铁道（railway）

【题目描述】

坐落于圣巢中心的泪水之城曾经是圣巢的首都，蓝湖的水从泪水之城的顶部不断渗透下来，形成了这里连绵不断的雨。

你在这里发现了 n 座失落的电车站，这些电车站之间用铁道连接着，形成了一条环线，方便起见将所有电车站从某个开始逆时针标号为第 $1, 2, 3 \ldots n$ 座电车站。n 段铁道连接着这些电车站，第 i 座电车站与第 $i \bmod n + 1$ 座电车站之间有一段长度为 l_i 的铁道。

每一段电车铁道都是单向的，不过由于年久失修，早已分辨不出每段铁道的方向。你准备重修这些铁道，并且可以在重修的时候独立地决定每一段铁道的方向（从 i 开向 $i \bmod n + 1$ 或者从 $i \bmod n + 1$ 开往 i）。

泪城是各路交通的中心，在旅途中你需要多次在泪城中移动，此时搭乘电车无疑是很方便的选择。你从先知处了解到，将来你会搭乘 m 次电车，第 i 次需要电车从第 s_i 座电车站驶向第 t_i 座电车站，当然电车在移动的过程中只能沿着铁道的方向。在第 i 次移动中，你携带了 w_i 单位重的货物，如果这次移动的距离为 d_i，那么需要花费的时间是 $w_i \times d_i$。

听了先知的话，请你决定应该如何规划每段铁道的方向，才能使得搭乘电车完成这 m 次移动的总时间花费最少？请输出最少的总时间，容易证明至少存在一种合法的规划方法。

【输入格式】

第一行两个正整数 n, m，含义见题目描述。

接下来 n 行每行一个正整数 l_i，表示连接第 i 和第 $i \bmod n + 1$ 座电车站的铁道长度。

接下来 m 行，每行三个整数 s_i, t_i, w_i 代表一次电车旅行的起点，终点和货物的重量。

【输出格式】

一行代表最少的总时间花费。

【样例 1 输入】

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

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

【样例 1 输出】

```
55
```

【样例 1 解释】

以下是一种可能的方案，每一段铁轨的方向分别是 $2 \to 1, 3 \to 2, 4 \to 3, 4 \to 5, 5 \to 6, 6 \to 1$。

第一次搭乘电车沿着 $4 \to 3 \to 2$，花费是 $(2+3) \times 5 = 25$。

第二次搭乘电车沿着 $4 \to 3 \to 2 \to 1$，花费是 $(1+2+3) \times 3 = 18$。

第三次搭乘电车沿着 $5 \to 6 \to 1$，花费是 $(5+1) \times 2 = 12$。

总共的花费是 55。

【样例 2】

见选手目录下 ***railway2.in*** 与 ***railway2.out***，该样例满足子任务一的性质。

【样例 3】

见选手目录下 ***railway3.in*** 与 ***railway3.out***，该样例满足子任务二的性质。

【样例 4】

见选手目录下 ***railway4.in*** 与 ***railway4.out***，该样例满足子任务三的性质。

【样例 5】

见选手目录下 ***railway5.in*** 与 ***railway5.out***，该样例满足子任务四的性质。

【样例 6】

见选手目录下 ***railway6.in*** 与 ***railway6.out***，该样例满足子任务五的性质。

【样例 7】

见选手目录下 ***railway7.in*** 与 ***railway7.out***，该样例满足子任务六的性质。

【样例 8】

见选手目录下 ***railway8.in*** 与 ***railway8.out***，该样例满足子任务七的性质。

【子任务】

对于全部的数据，$2 \le n \le 5 \times 10^5, 1 \le m \le 5 \times 10^5, 1 \le l_i, w_i \le 10^3, s_i \ne t_i$。

子任务编号	分值	$n \le$	$m \le$	特殊性质
1	15	20	20	无
2	15	5×10^5	18	无
3	10	300	300	无
4	15	3000	3000	无
5	15	5×10^5	5×10^5	$\forall i, s_i < t_i$
6	15	10^5	10^5	无
7	15	5×10^5	5×10^5	无

走过安眠地的花丛（death）

【题目描述】

穿过安息之地，来到了一片山崖，这里长满了彼岸花，鲜红的，就像正在燃烧一样。花丛中总是隐藏着不可知的危险，因此来到这里的人只能沿着花丛间的小路行进。

人们的目的地是最远处，离死亡最近的地方，去迎接终结，亦是新生。而你是掌管死亡的神，你创造了这片花丛，也创造了这些小路。

你创造的小路形成了 n 个路口，你将他们依次编号为 $1,2,3\dots n$，起点是第 1 个路口，而终点是第 n 个路口。第 i 条小路从编号为 u_i 的路口走向编号为 v_i 的路口，距离为 l_i 个单位长度，由于去往死亡的路上不允许回头，故可以认为每条小路都是单向的，只能从 u_i 走向 v_i。

一个人从起点开始时认为是 0 时刻，每走一个单位长度花费的时间是一个单位时间，中间不能进行停留或者回头，假设其到达终点的时刻是 t 时刻。不同的人可能会选择不同的行进道路，因此最终到达终点的时刻是不尽相同的，而身为神明的你总是希望人们在你指定的若干时刻里到达终点。具体来说你有一个长度为 k 的时刻表，里面有 k 个时刻 $a_1,a_2\dots,a_k$，你希望所有人们可能到达的时刻均在这个时刻表里，并且人们不可能到达的时刻都不在这个时刻表里。

如今你掌管死亡，应该如何设计这片花丛间的道路，才能满足上面的要求呢？

由于创造花费神力，因此你希望路口的个数 n 比较少（并不要求最少，具体见评分细则），并且每条道路的长度均是 $[0,2000]$ 的一个整数。

【输入格式】

第一行一个整数 k 代表时刻表的大小。

第二行 k 个整数，$a_1,a_2\dots a_k$。

【输出格式】

第一行两个整数 n,m，代表你构造的图的点数和边数。

接下来 m 行，每行三个整数 u_i,v_i,w_i 代表一条边。

你应当保证：

- $1\le n\le 100, 1\le m\le \frac{n(n-1)}{2}$
- $1\le u_i<v_i\le n$
- $0\le w_i\le 2000$。
- $\forall 1\le i<j\le n,(u_i,v_i)\ne(u_j,v_j)$

【样例 1 输入】

```
4
1 2 4 5
```

【样例 1 输出】

```
4 6
1 2 2
1 3 3
1 4 1
2 3 2
2 4 0
3 4 1
```

【评分细则】

本题有若干测试点，你的得分是每个测试点得分的最小值。

对于每个测试点，如果你给出的 G 不合法则得 0 分；否则设 n 为你给出的图的点数，你的得分 p 为：

- $100 < n$，$p = 0$。
- $92 < n \le 100$，$p = 5 + \frac{100-n}{8} \times 5$。
- $80 < n \le 92$，$p = 10 + \frac{92-n}{12} \times 10$。
- $69 < n \le 80$，$p = 20 + \frac{80-n}{11} \times 15$。
- $60 < n \le 69$，$p = 35 + \frac{69-n}{9} \times 30$。
- $53 < n \le 60$，$p = 65 + \frac{60-n}{7} \times 35$。
- $n \le 53$，$p = 100$。

对于 100% 的数据，保证 $1 \le m \le 2000, 1 \le a_1 < a_2 < a_3 \cdots < a_m \le 2000$。

昔在、今在、永在的题目（problem）

【题目背景】

时光荏苒，你坐在 NOI6202 的考场上，看着眼前的这一道题目，无力感席卷了全身，你知道，一切旅程将要到达终点了。

“为什么，为什么当初我没有做那道题目……”

闭上眼，回想起自己曾经在 OI 赛场上挥洒汗水的场景。

眼泪还是流了出来。你不甘心，你觉得你还可以做得更好。

你突然惊醒，发现自己回到了省队集训的那一天。

之前经历的一切仿佛是一场梦，却又那么真实。

你意识到，这一次，你还有机会。

你决定，这一次，让 OI 生涯不留遗憾。

你充满了决心。

【题目背景 2】

这里本来是一道很有意思的题目，但是因【数据删除】所以没有了。

【题目描述】

一年一度的综艺节目《中国好码龙》又开始了，来自全国各地的小码龙们都来参加竞赛了。本次节目一共邀请了 m 位导师。方便起见，接下来用 $1 \sim m$ 的编号来称呼这 m 位导师。

现在比赛已经进行到了战队赛阶段，一共还剩下有 n 位选手，我们把这些选手从 $1 \sim n$ 编号，编号为 i 的选手属于编号为 a_i 的导师的战队，可能有的导师的战队里没有龙。

现在进入了激动人心的战队 PK 环节，所有选手按照编号顺序从左到右在舞台上排成一排，选手们要两两进行 PK 赛。接下来对于所有选手，如果其编号为 i，他可以找一位属于编号为 $a_i \bmod m + 1$ 的导师的战队，且其编号 j 满足 $j > i$ 的选手进行 PK；当然也可以成为某个选手的对手，不过每名选手只能成为一名选手的对手，这一过程后选手两人一组分成了若干组。

但是这样会导致部分选手无法选到对手，因此为了节目的进行，主办方找到了一些选手通过重金劝说其离开，不妨认为让一名选手退赛需要花费 1 单位的龙币。

你名叫奶龙，是《中国好码龙》的总策划，你需要抉择让哪些选手离开可以让剩下的选手存在至少一种符合要求的配对方案，并且你希望花费的龙币尽可能少。

不过这个消息走漏了，因此在接下来的 q 天里，第 i 天编号为 x_i 的选手投靠了编号为 y_i 的战队，你需要在所有最开始以及每一天结束后都求出最少花费多少单位龙币可以打成目标?

更换战队是永久的。

【输入格式】

第一行三个整数 n, m, q 分别代表选手数量，导师数量，以及询问的天数。
第二行 n 个整数，第 i 个整数 a_i 代表选手 i 的战队。
接下来 q 行，每行两个整数 x_i, y_i 代表编号为 x_i 的选手投靠了编号为 y_i 的战队。

【输出格式】

共 $q+1$ 行，每行一个非负整数代表当天最少花费多少单位龙币可以打成目标?

【样例 1 输入】

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

【样例 1 输出】

```
1
1
1
1
3
3
```

【样例 1 解释】

一开始，只需要请 5 号选手离开，就能让 1, 2 和 3, 4 号选手完成配对。
第一次操作后，请 1 号选手离开，2, 5 和 3, 4 号选手完成配对。

【样例 2】

见选手目录下 ***problem2.in*** 与 ***problem2.ans***，该样例满足子任务 1 的性质。

【样例 3】

见选手目录下 ***problem3.in*** 与 ***problem3.ans***，该样例满足子任务 2 的性质。

【样例 4】

见选手目录下 ***problem4.in*** 与 ***problem4.ans***，该样例满足子任务 3 的性质。

【样例 5】

见选手目录下 ***problem5.in*** 与 ***problem5.ans***，该样例满足子任务 4 的性质。

【样例 6】

见选手目录下 ***problem6.in*** 与 ***problem6.ans***，该样例满足子任务 5 的性质。

【样例 7】

见选手目录下 ***problem7.in*** 与 ***problem7.ans***，该样例满足子任务 6 的性质。

【样例 8】

见选手目录下 ***problem8.in*** 与 ***problem8.ans***，该样例满足子任务 7 的性质。

【数据规模】

对于 100% 的数据，$1 \le n, m, q \le 3 \times 10^5, 1 \le a_i, y_i \le m, 1 \le x_i \le n, m \ge 2, n \ge 2$

子任务编号	$n \le$	$m \le$	$q \le$	特殊性质	分值
1	30	30	100	无	10
2	500	500	500	无	15
3	5000	5000	5000	无	15
4	10^5	10^5	10^5	无	20
5	3×10^5	2	3×10^5	无	10
6	3×10^5	3×10^5	3×10^5	$m \bmod 2 = 0$	10
7	3×10^5	3×10^5	3×10^5	无	20