

爬山 (climb)

【题目描述】

在一片神秘的大陆上，居住着一位名叫 `pty` 的旅行者。`pty` 最喜欢的活动之一就是攀登高山，他计划从这片大陆的 **1 号山** 出发，最终抵达 **n 号山**。

这片大陆共有 n 座山和 m 条道路，每条道路双向连接两座山。

`pty` 初始拥有 k 点体力。在攀登的过程中：

- 若海拔上升 1m ，`pty` 会消耗 1 点体力；
- 若海拔下降 1m ，`pty` 会恢复 1 点体力。

然而，`pty` 的体力在任何时候都不能为负数。为了顺利到达终点，他可以事先请铁匠 `cch` 帮忙降低某些山的高度。将一座山降低 l 米需要花费 $l \times l$ 的金币，并且每座山至多只能被降低一次。由于 `pty` 已经站在 1 号山上，这座山的高度不能被修改。

`pty` 花费的总代价 = 降低山峰的代价 + 行走路径的总长度。

请你计算：`pty` 从 **1 号山** 到 **n 号山**，所需的最小总代价。

【输入格式】

从文件 `climb.in` 中读入数据。

第一行包含三个整数 n, m, k 。

第二行包含 n 个整数，第 i 个整数 h_i 表示第 i 座山的高度。

接下来 m 行，每行包含三个整数 x, y, z ，表示 x, y 之间存在一条长度为 z 的双向道路。

保证 1 号山与 n 号山连通。

【输出格式】

输出到文件 `climb.out` 中。

输出一个整数，表示最小总代价。

【样例 1 输入】

```
4 5 1
1 2 3 4
1 2 1
1 3 1
1 4 100
2 4 1
3 4 1
```

【样例 1 输出】

6

【样例 2】

见选手目录下的 *climb/climb2.in* 与 *climb/climb2.ans*。

【样例 3】

见选手目录下的 *climb/climb3.in* 与 *climb/climb3.ans*。

【数据范围】

对于所有测试数据，保证：

- $1 \leq n, k, h_i, z \leq 10^5$;
- $1 \leq m \leq 2 \times 10^5$;
- $1 \leq x, y \leq n, x \neq y$ 。

测试点	n	m	k, h_i 范围
1 ~ 5	≤ 200	≤ 200	≤ 200
6 ~ 10	$\leq 5,000$	$\leq 5,000$	$\leq 5,000$
11 ~ 15	$\leq 70,000$	$\leq 70,000$	$\leq 10^5$
16 ~ 20	$\leq 10^5$	$\leq 200,000$	$\leq 10^5$