

机器人吃东西 (eat)

【题目描述】

dottle 喜欢吃东西。

他有 n 个食物，每个食物都有属性 a_i 。

每一天，dottle 必须选择一个非空区间 $[l, r]$ ，满足 $a_l \sim a_r$ 所有数的按位与为 0，然后吃掉这个区间的所有食物。吃掉之后左边和右边合并起来。

比如，假设初始序列为 $\{3, 5, 4, 1, 4, 7\}$ ，dottle 可以选择 $[2, 4]$ 这个区间吃掉，因为 $5 \& 4 \& 1 = 0$ ，在这之后，这个序列变成 $\{3, 4, 7\}$ 。

现在 dottle 想知道，他最多能进行多少次吃操作？

【输入格式】

从文件 *eat.in* 中读入数据。

第一行一个正整数 n 。

之后一行 n 个非负整数，第 i 个为 a_i 。

【输出格式】

输出到文件 *eat.out* 中。

输出一行一个整数，表示答案。

【样例输入 1】

```
5
1 5 7 2 6
```

【样例输出 1】

```
2
```

【样例解释 1】

第一次吃掉 $[2, 4]$ 这个区间，序列变为 $[1, 6]$

第二次吃掉 $[1, 2]$ 这个区间，序列变为空。

最终次数为 2。

注意你不需要让最终的序列为空。

【样例 2】

见下发文件中的 *eat2.in/out*。

该样例满足测试点 4 ~ 8 的限制。

【样例 3】

见下发文件中的 eat3.in/out。

该样例满足测试点 9 ~ 11 的限制。

【样例 4】

见下发文件中的 eat4.in/out。

该样例满足测试点 14 ~ 17 的限制。

【数据范围与提示】

对于所有数据，满足 $1 \leq n \leq 63, 0 \leq a_i \leq 63$ 。

每个测试点的具体限制见下表：

测试点编号	n	a_i	特殊性质
1 ~ 3	≤ 5	≤ 15	无
4 ~ 8	≤ 15	≤ 63	
9 ~ 11	≤ 31	≤ 31	
12 ~ 13		≤ 1	
14 ~ 17	≤ 63	≤ 63	保证所有 a_i 可以表示为 2^k 的形式，其中 $k \geq 0$
18 ~ 20			无

机器人的积木 (block)

【题目描述】

dottle 在搭积木。

有 n 堆积木，第 i 堆积木的高度为 h_i ，dottle 想让积木堆尽量平均，所以他定义一个状态的不优美度为 $\sum_{i=1}^{n-1} |h_i - h_{i+1}| + h_1 + h_n$ 。

dottle 只能执行一种操作：将一堆积木的其中一块拿走，也就是选定某个 $h_x > 0$ 的 x ，令 $h_x = h_x - 1$ 。

q 次询问，每次给出一个 X ，求使用不超过 X 次操作后不优美度的最小值。

【输入格式】

从文件 `block.in` 中读入数据。

第一行两个正整数 n, q 。

接下来一行 n 个非负整数，表示初始的 h 序列。

之后 q 行，每行一个非负整数，表示询问的 X 。

【输出格式】

输出到文件 `block.out` 中。

对于每组询问，输出一行一个数表示最终答案。

【样例输入 1】

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

【样例输出 1】

```
12
10
```

8
8
6
6
6
6
6
4

【样例 2】

见下发文件中的 block2.in/out
该样例满足测试点 4 ~ 9 的限制。

【样例 3】

见下发文件中的 block3.in/out
该样例满足测试点 10 ~ 13 的限制。

【样例 4】

见下发文件中的 block4.in/out
该样例满足测试点 19 ~ 21 的限制。

【数据范围与提示】

对于所有测试点： $2 \leq n, q \leq 5 \times 10^5, 0 \leq a_i \leq 10^{12}, 0 \leq X \leq 10^{18}$ 。
每个测试点的具体限制见下表：

测试点编号	n	q	a_i	X
1 ~ 3	≤ 5	≤ 5	≤ 5	≤ 5
4 ~ 9	≤ 500	≤ 10	≤ 50	$\leq 5 \times 10^3$
10 ~ 13	$\leq 10^3$	$\leq 5 \times 10^5$	$\leq 10^{12}$	$\leq 10^{15}$
14 ~ 15	$\leq 10^5$		≤ 1	$\leq 10^6$
16 ~ 18		≤ 1		
19 ~ 21		≤ 100		
22 ~ 25		$\leq 5 \times 10^5$	$\leq 10^{12}$	$\leq 10^{18}$

机器人填数 (tree)

【题目描述】

dottle 得到了一棵 n 个点的以 1 为根的有根树，他想要在树上每一个点中填一个 $1 \sim m$ 的颜色。

同时给定序列 f ，其中 f_i 表示限制以 i 为根的子树中恰好有 f_i 种不同的颜色，若 $f_i = -1$ ，则表示没有任何限制。

求填颜色的方案数，对 $10^9 + 7$ 取模，保证答案在取模之前不为 0。

【输入格式】

从文件 `tree.in` 中读入数据。

第一行两个正整数 n, m 。

接下来 $n - 1$ 行，每行两个数 x, y ，表示树中的一条边。

接下来一行 n 个数，表示 $f_1 \sim f_n$ 。

【输出格式】

输出到文件 `tree.out` 中。

输出一行一个数，表示你的答案，对 $10^9 + 7$ 取模。

【样例输入 1】

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

【样例输出 1】

```
48
```

【样例 2】

见下发文件中的 `tree2.in/out`。

该样例满足测试点 3 ~ 5 的限制。

【样例 3】

见下发文件中的 `tree3.in/out`。

该样例满足测试点 10 ~ 11 的限制。

【样例 4】

见选手目录下的 tree4.in/out。

该样例满足测试点 15 ~ 18 的限制。

【数据范围与提示】

对于所有测试点： $2 \leq n \leq 10^5$, $1 \leq m \leq 10^5$, $-1 \leq f_i \leq m$ ，保证答案在取模之前不为 0。
每个测试点的具体限制见下表：

测试点编号	n	m	特殊性质
1 ~ 2	≤ 5	≤ 5	无
3 ~ 5	≤ 15	≤ 15	A
6 ~ 7	≤ 50	≤ 50	
8 ~ 9	$\leq 2 \times 10^3$	≤ 100	
10 ~ 11			无
12 ~ 13	$\leq 10^5$	≤ 10	A
14		≤ 10	无
15 ~ 18		≤ 100	A
19		≤ 100	无
20 ~ 24		$\leq 10^5$	A
25		$\leq 10^5$	无

操作 (operation)

【题目描述】

给定一个长度为 n 的序列 a 。

您有 $m + n$ 种可执行的操作。

前 m 种的第 i 种形如 X_i, L_i, R_i, W_i 表示您可以把 a_{X_i} 减 1, 再把 $a_{L_i} \sim a_{R_i}$ 加 1, 这个操作会有 W_i 的代价。

后 n 种的第 i 种仅含一个数 b_i , 表示直接令 $a_i - 1$, 花费 b_i 的代价, 特别的, 若 $b_i == -1$, 代表这个操作是不可执行的。

求最小的代价, 让所有的 a_i 变为 0, 保证若存在合法的操作方案, 代价最小的操作方案花费不超过 2×10^{18} 。

【输入格式】

从文件 `operation.in` 中读入数据。

第一行两个整数 n, m 。

之后一行 n 个整数, 第 i 个数为 a_i 。

之后 m 行每行四个整数 X_i, L_i, R_i, W_i 。

之后一行 n 个整数, 第 i 个数为 b_i 。

【输出格式】

输出到文件 `operation.out` 中。

若可以让所有 a_i 变为 0, 输出最小的代价。

否则输出 -1 。

【样例输入 1】

```
4 2
0 1 0 0
2 3 4 5
2 1 1 0
10 -1 2 2
```

【样例输出 1】

```
9
```

【样例 2】

见下发文件中的 `operation2.in/out`。

该样例满足测试点 1 ~ 2 的限制。

【样例 3】

见下发文件中的 operation3.in/out。
该样例满足测试点 3 ~ 7 的限制。

【样例 4】

见下发文件中的 operation4.in/out。
该样例满足测试点 8 ~ 11 的限制。

【数据范围与提示】

对于所有数据，满足 $1 \leq n, m \leq 4 \times 10^5$ ， $0 \leq W_i \leq 10^9$ ， $0 \leq a_i \leq 10^3$ ， $-1 \leq b_i \leq 10^9$ ， $L_i \leq R_i$ 。

每个测试点的具体限制见下表：

测试点编号	n	m	特殊性质
1 ~ 2	≤ 2	≤ 514	无
3 ~ 7	$\leq 10^3$	$\leq 10^3$	
8 ~ 11	$\leq 10^5$	$\leq 10^5$	$L_i = R_i$
12 ~ 14			$W_i = 0, b_i \leq 0$
15 ~ 18			无
19 ~ 20	$\leq 4 \times 10^5$	$\leq 4 \times 10^5$	