CSP-S2022 模拟赛

gyh20

题目名称	机器人游戏	机器人和城市 2	机器人的积木	机器人操作
题目类型	传统型	传统型	传统型	传统型
输入文件名	game.in	city.in	block.in	operation.in
输出文件名	game.out	city.out	block.out	operation.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	2.0 秒
内存限制	512 MB	512 MB	512 MB	512 MB
测试点数目	10	10	25	20
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	game.cpp	city.cpp	number.cpp	operation.cpp

编译选项

对于 C++ 语言	-lm -O2 -std=c++14

注意事项

1. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
2. C++ 中函数 `main()` 的返回值类型必须是 `int`，值必须为 0。
3. 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分隔。
4. 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。
5. 原则上，每个测试点时限应为标准程序在该测试点上的运行时间的 2 倍及以上。
6. 每道题的时间限制、编译命令、是否开启文件输入输出等信息，在赛时均有可能变动，请各位选手以赛时通知为准。
7. AK 了不要声张，闷声发大财。

机器人游戏（game）

【题目描述】

dottle 正在玩经典的猜数游戏。

给定 n, m，初始有两个变量 $l = 1, r = n$，同时有一个神秘数字 x，dottle 每次可以猜一个 $[l, r]$ 之间的整数 y。

若 $y = x$，则游戏立刻结束。

若 $y < x$，则令 $r = y - 1$，游戏继续。

若 $y > x$，则令 $l = y + 1$，游戏继续。

每次猜测会花费 1 的代价。

dottle 会玩很多局游戏，每一次神秘数字会从 $[1, n]$ 中的整数中随机选取，每一次 dottle 会记录游戏的总次数和总代价，特别的是，若一局游戏的代价 $> m$，他会假装他没有玩这局游戏，也就是说，当前游戏不会使总次数或者总代价发生变化。

你需要帮助 dottle 找到一个策略，使得最终的期望的平均次数（总代价/总次数）最小，并输出这个值，相对误差或绝对误差不超过 10^{-6} 视为正确。

【输入格式】

从文件 ***game.in*** 中读入数据。

一行两个正整数 n, m。

【输出格式】

输出到文件 ***game.out*** 中。

输出一行一个数 ans，表示最终的答案。

【样例输入 1】

```
3 2
```

【样例输出 1】

```
1.5
```

【样例解释 1】

考虑一下两种策略：

1. 第一次猜 2，若猜中则次数为 1，结束，否则下一次一定猜中，这样的期望是 $\frac{1+2\times 2}{3}=\frac{5}{3}$。

2. 第一次猜 1，若猜中则次数为 1，结束，否则第二次猜 2，若猜中则次数为 2，如果没有猜中由于次数 >2 则不会被记录，所以不会影响，这样的期望是 $\frac{1+2}{2}=\frac{3}{2}$。

后一种策略显然更优。

【样例输入 2】

```
114514 1
```

【样例输出 2】

```
1
```

【样例解释 2】

无论使用什么策略。只要次数不是 1 都不会被统计入答案。最终的次数一定为 1。

【数据范围与提示】

每个测试点的具体限制见下表：

测试点编号	n	m
$1\sim 2$	≤ 10	$\le n$
$3\sim 4$	≤ 50	
5	≤ 500	
6	$\le 5\times 10^3$	
7	$\le 10^6$	
$8\sim 9$	$\le 10^9$	$=n$
10		$\le n$

机器人和城市 2（city）

【题目描述】

dottle 也生活在城市里。

城市可以看做一个 n 个点 m 条边的无向连通图。

每条边连接了两个端点，并有两个属性，w_i, h_i，表示路径的长度为 1000000000^{w_i}，以及 dottle 对这条路径的喜爱度为 h_i，保证 w_i, h_i 是非负整数，且保证所有 w_i 互不相同。

定义 dottle 对两个点 x, y 的喜爱度 $f(x, y)$ 为：在所有 x 到 y 的长度和最小的路径中，喜爱度之和最大的一条。

dottle 想找到一个排列 p，使得 $\sum_{i=1}^{n-1} f(p_i, p_{i+1}) + f(p_1, p_n)$ 最大。

【输入格式】

从文件 ***city.in*** 中读入数据。

第一行两个整数 n, m。

之后 m 行，每行四个整数 u, v, w, h，表示该边连接 u, v，属性为 (w, h)。

【输出格式】

输出到文件 ***city.out*** 中。

输出一行一个整数，表示答案。

【样例输入 1】

```
4 6
1 2 4 9
1 3 6 12
4 1 2 3
4 2 3 4
2 3 5 1
4 3 1 4
```

【样例输出 1】

```
22
```

【样例 2】

见下发文件中的 city2.in/out。

该样例满足测试点 2 ~ 3 的限制。

【样例 3】

见下发文件中的 city3.in/out。
该样例满足测试点 $4 \sim 5$ 的限制。

【样例 4】

见下发文件中的 city4.in/out。
该样例满足测试点 $9 \sim 10$ 的限制。

【数据范围与提示】

对于所有数据，满足 $1 \le n \le 10^5, 1 \le m \le 2 \times 10^5, 0 \le w_i, h_i \le 10^9$ 且所有 w_i 互不相同。
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	≤ 4	≤ 6	无
$2 \sim 3$	≤ 10	≤ 20	
$4 \sim 5$	≤ 50	$= n-1$	
$6 \sim 7$	$\le 10^5$		第 i 条边连接 i 与 $i+1$
8			第 i 条边连接 1 与 $i+1$
$9 \sim 10$		$\le 2 \times 10^5$	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 $4 \sim 9$ 的限制。

【样例 3】

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

【样例 4】

见下发文件中的 block4.in/out
该样例满足测试点 $19 \sim 21$ 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：

<table>
<tr><th>测试点编号</th><th>n</th><th>q</th><th>a_i</th><th>X</th></tr>
<tr><td>$1 \sim 3$</td><td>≤ 5</td><td>≤ 5</td><td>≤ 5</td><td>≤ 5</td></tr>
<tr><td>$4 \sim 9$</td><td>≤ 500</td><td>≤ 10</td><td>≤ 50</td><td>$\leq 5 \times 10^3$</td></tr>
<tr><td>$10 \sim 13$</td><td>$\leq 10^3$</td><td rowspan="2">$\leq 5 \times 10^5$</td><td>$\leq 10^{12}$</td><td>$\leq 10^{15}$</td></tr>
<tr><td>$14 \sim 15$</td><td rowspan="3">$\leq 10^5$</td><td>≤ 1</td><td rowspan="3">$\leq 10^6$</td></tr>
<tr><td>$16 \sim 18$</td><td>≤ 1</td><td rowspan="2">≤ 100</td></tr>
<tr><td>$19 \sim 21$</td><td rowspan="2">$\leq 5 \times 10^5$</td></tr>
<tr><td>$22 \sim 25$</td><td>$\leq 5 \times 10^5$</td><td>$\leq 10^{12}$</td><td>$\leq 10^{18}$</td></tr>
</table>

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 $1 \sim 2$ 的限制。

【样例 3】

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

【样例 4】

见下发文件中的 operation4.in/out。
该样例满足测试点 $8 \sim 11$ 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
$1 \sim 2$	≤ 2	≤ 514	无
$3 \sim 7$	$\leq 10^3$	$\leq 10^3$	
$8 \sim 11$	$\leq 10^5$	$\leq 10^5$	$L_i = R_i$
$12 \sim 14$			$W_i = 0, b_i \leq 0$
$15 \sim 18$			无
$19 \sim 20$	$\leq 4 \times 10^5$	$\leq 4 \times 10^5$	