

# NOI2025 day1 模拟赛

题目名称	图上的游戏	修理	人员调度 2
题目类型	传统型	传统型	传统型
目录	game	repair	transfer
可执行文件名	game	repair	transfer
输入文件名	game.in	repair.in	transfer.in
输出文件名	game.out	repair.out	transfer.out
每个测试点时限	4.0 秒	1.5 秒	5.0 秒
内存限制	512 MiB	1024 MiB	1024 MiB
测试点数目	25	20	25
测试点是否等分	是	是	是

## 注意事项与提醒（请选手务必仔细阅读）

1. 输入输出文件名必须使用英文小写。
2. 选手提交的源程序必须存放在**已建立好的**，且**带有样例文件和下发文件**的文件夹中，文件夹名称与对应试题英文名一致。
3. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
4. C++ 中函数 `main()` 的返回值类型必须是 `int`，值必须为 `0`。
5. 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**。
6. 程序可使用的栈空间大小与该题内存空间限制一致。
7. 在终端中执行命令 `ulimit -s unlimited` 可将当前终端下的栈空间限制放大，但你使用的栈空间大小不应超过题目限制。
8. 每道题目所提交的**代码文件大小限制为 100KB**。
9. 若无特殊说明，输入文件与输出文件中同一行的相邻整数均使用一个空格分隔。
10. 输入文件中可能存在行末空格，请选手使用更完善的读入方式（例如 `scanf` 函数）避免出错。
11. 直接复制 PDF 题面中的多行样例，数据将带有行号，建议选手直接使用对应目录下的样例文件进行测试。
12. 使用 `std::deque` 等 STL 容器时，请注意其内存空间消耗。
13. 请务必使用题面中规定的编译参数，保证你的程序在本机能够通过编译。此外**不允许在程序中手动开启其他编译选项**，一经发现，本题成绩以 0 分处理。
14. **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。**

## 图上的游戏 (game)

### 【题目描述】

我喜欢你, E\_Space, 所以这是一道博弈题。

小 A 与小 B 喜欢图上的游戏, 现在有一个  $n$  个点的有向图  $G$ , 最初图上没有边。在接下来  $m$  个时刻, 每个时刻会发生下列事件:

- 随机选取  $u, v \in [1, n]$ , 在图  $G$  上连接一条从  $u \rightarrow v$  的有向边。

小 A 和小 B 喜欢黑白两种颜色, 最初第  $i$  个点有颜色  $c_i \in \{0, 1, 2\}$ , 如果  $c_i = 0$  就是黑色,  $c_i = 1$  就是白色,  $c_i = 2$  表示该点颜色未确定。

小 A 和小 B 会在  $m$  个时刻后的图  $G$  上做游戏, 小 A 与小 B 会轮流选择图  $G$  上一个还未染色的点进行染色, 其中小 A 先手。小 A 和小 B 认为黑白相间的边才是美丽的, 但是边存在方向。小 A 获得的分数是满足  $u \rightarrow v$  且  $c_u = 0, c_v = 1$  的边的条数, 小 B 获得的分数是满足  $u \rightarrow v$  且  $c_u = 1, c_v = 0$  的边的条数, 令  $Z$  为小 A 的分数减去小 B 的分数, 小 A 想要使  $Z$  尽量大, 而小 B 想要使  $Z$  尽量小, 若双方都足够聪明, 最终  $Z$  的值会是多少?

小 A 和小 B 想要知道, 对于所有可能的  $n^{2m}$  种情况, 最终  $Z$  的对  $10^9 + 7$  取模后的值为多少。注意, 这里认为两种情况不同, 当前仅当存在某个时刻加入的有向边不同, 同时, 图  $G$  中可能存在重边和自环。

但是观战者小 C 认为这太简单了, 小 C 想要知道, 对于所有  $1 \leq N \leq n, 1 \leq M \leq m$ , 若只关注图  $G$  的前  $N$  个点 (即忽略后  $n - N$  个点, 图  $G$  上只有  $N$  个点) 且  $m = M$  时上述问题对应的答案。

### 【输入格式】

从文件 `game.in` 中读入数据。

第一行两个整数  $n, m$ , 分别表示图  $G$  的点数以及时刻数。

接下来一行包含  $n$  个整数, 第  $i$  个整数表示  $c_i$ 。

### 【输出格式】

输出到文件 `game.out` 中。

共  $n$  行, 每行包含  $m$  个整数, 第  $i$  行第  $j$  个整数表示当  $N = i, M = j$  时对应的答案。

**【样例 1 输入】**

```
1 3 2
2 0 1 2
```

**【样例 1 输出】**

```
1 0 0
2 0 0
3 2 28
```

见选手目录下的 *game/game1.in* 与 *game/game1.ans*。

**【样例 2】**

见选手目录下的 *game/game2.in* 与 *game/game2.ans*，该组样例满足测试点 7 的条件。

**【样例 3】**

见选手目录下的 *game/game3.in* 与 *game/game3.ans*，该组样例满足测试点 12 的条件。

**【样例 4】**

见选手目录下的 *game/game4.in* 与 *game/game4.ans*，该组样例满足测试点 18 的条件。

**【样例 5】**

见选手目录下的 *game/game5.in* 与 *game/game5.ans*，该组样例满足测试点 22 的条件。

**【数据范围】**

对于 100% 的数据， $1 \leq n, m \leq 50$ ， $c_i \in \{0, 1, 2\}$ 。

测试点编号	$n$	$m$	特殊限制
1	$\leq 50$	$\leq 50$	A
2 ~ 3	$\leq 50$	$= 1$	无
4 ~ 6	$\leq 50$	$= 2$	无
7 ~ 9	$\leq 5$	$\leq 5$	无
10 ~ 11	$\leq 8$	$\leq 8$	无
12 ~ 13	$\leq 15$	$\leq 15$	无
14 ~ 17	$\leq 25$	$\leq 25$	无
18 ~ 21	$\leq 50$	$\leq 50$	B
22 ~ 23	$\leq 35$	$\leq 35$	无
24 ~ 25	$\leq 50$	$\leq 50$	无

**特殊性质 A:**  $\forall i \in [1, n], c_i \in \{0, 1\}$ 。

**特殊性质 B:** 保证  $c_i = 2$  的个数不超过 1。

## 修理 (repair)

### 【题目描述】

又到了小机器人维修时间！

由于工作损耗，小机器人可能会偏移了正常的行为设定，我们用  $b_i$  表示每个小机器人的偏移值。你的任务是将所有小机器人恢复正常（即将所有偏移值变为 0）。你有一把脉冲枪来完成这项任务，脉冲枪初始脉冲值  $x = 0$ ，单位时间内，你可以进行如下操作之一：

- 将  $x$  增加 1。
- 对一个小机器人进行施加异或脉冲，即选择一个  $i$ ，令  $b_i \leftarrow b_i \oplus x$ 。

其中  $\oplus$  表示二进制下的异或运算。

现在，给定长度为  $n$  的序列  $a_i$  表示小机器人的偏移值，多次询问修理区间  $[l, r]$  中所有小机器人所需花费的最少时间。

### 【输入格式】

从文件 *repair.in* 中读入数据。

第一行三个整数  $n, Q, t$ ，分别表示小机器人数量、询问次数、是否强制在线。

第二行共  $n$  个正整数，分别表示每个小机器人的偏移值  $a_i$ 。

接下来  $Q$  行，每行两个整数  $l'_i, r'_i$  代表一次询问。当  $t = 1$  时  $l_i = l'_i, r_i = r'_i$ 。当  $t = 2$  时， $l_i = \min(l'_i \oplus lst, r'_i \oplus lst), r_i = \max(l'_i \oplus lst, r'_i \oplus lst)$ ，其中  $lst$  表示上一次询问的答案（初始为 0）。

### 【输出格式】

输出到文件 *repair.out* 中。

共  $Q$  行，第  $i$  行表示第  $i$  次修理所需的最少时间。

### 【样例 1 输入】

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

**【样例 1 输出】**

```
1 9
2 11
3 12
4 14
5 12
6 6
```

见选手目录下的 *repair/repair1.in* 与 *repair/repair1.ans*。

**【样例 2】**

见选手目录下的 *repair/repair2.in* 与 *repair/repair2.ans*，该组样例满足测试点 1 的条件。

**【样例 3】**

见选手目录下的 *repair/repair3.in* 与 *repair/repair3.ans*，该组样例满足测试点 3 的条件。

**【样例 4】**

见选手目录下的 *repair/repair4.in* 与 *repair/repair4.ans*，该组样例满足测试点 5 的条件。

**【样例 5】**

见选手目录下的 *repair/repair5.in* 与 *repair/repair5.ans*，该组样例满足测试点 7 的条件。

**【样例 6】**

见选手目录下的 *repair/repair6.in* 与 *repair/repair6.ans*，该组样例满足测试点 9 的条件。

**【样例 7】**

见选手目录下的 *repair/repair7.in* 与 *repair/repair7.ans*，该组样例满足测试点 11 的条件。

**【样例 8】**

见选手目录下的 *repair/repair8.in* 与 *repair/repair8.ans*, 该组样例满足测试点 14 的条件。

### 【样例 9】

见选手目录下的 *repair/repair9.in* 与 *repair/repair9.ans*, 该组样例满足测试点 17 的条件。

### 【数据范围】

对于 100% 的数据,  $1 \leq n, Q \leq 2 \times 10^5, 1 \leq l_i \leq r_i \leq n, 1 \leq a_i < 2^{60}$ 。

测试点编号	$n$	$Q$	$t$	特殊限制
1 ~ 2	$\leq 10^3$	$\leq 10^3$	1	无
3 ~ 4	$\leq 2 \times 10^5$	$\leq 2 \times 10^5$	1	特殊性质 A
5 ~ 6	$\leq 2 \times 10^5$	$\leq 2 \times 10^5$	1	特殊性质 B
7 ~ 8	$\leq 5 \times 10^4$	$\leq 5 \times 10^4$	1	无
9 ~ 10	$\leq 2 \times 10^5$	$\leq 2 \times 10^5$	1	无
11 ~ 13	$\leq 5 \times 10^4$	$\leq 5 \times 10^4$	2	无
14 ~ 16	$\leq 10^5$	$\leq 10^5$	2	无
17 ~ 20	$\leq 2 \times 10^5$	$\leq 2 \times 10^5$	2	无

特殊性质 A:  $\forall i \in [1, n), a_i \leq a_{i+1}$ 。

特殊性质 B:  $\forall i \in [1, Q], l_i = 1, r_i = i$ 。

## 人员调度 2 (transfer)

### 【题目描述】

两年以前,小 A 在省选 day1 场上遇到了《人员调度》此题,并精准识别出是 LOJ 黄金矿工的弱化版,可惜小 A 由于懒惰,没有去做该题,只能遗憾离场。

.....

小 A 现在是一家公司的老板,该公司共有  $n$  个员工,以及  $n$  个岗位。第  $i$  个员工的能力值为  $a_i$ ,第  $i$  个岗位的要求值为  $b_i$ ,第  $i$  个员工就职第  $j$  个岗位对公司产生的基础收益为  $a_i + b_j + (a_i \oplus b_j)$ 。其中  $\oplus$  表示二进制下的异或运算。

同时小 A 发现特定的员工就职特定的岗位会产生额外的效益。小 A 会给出  $m$  条信息,每条信息形如  $(x, y, w)$ ,即第  $x$  个员工就职第  $y$  个岗位会产生  $w$  的额外收益。

小 A 给出一个参数  $K$ ,他想要知道,对于  $1 \leq k \leq K$ ,若恰好有  $k$  个员工进行就职,产生的总收益(基础收益 + 额外收益)最大和为多少?

### 【输入格式】

从文件 `transfer.in` 中读入数据。

第一行三个整数  $n, m, K$ ,分别表示员工数/岗位数以及信息的条数,所给的参数。

第二行包含  $n$  个整数,第  $i$  个整数表示  $a_i$ 。

第三行包含  $n$  个整数,第  $i$  个整数表示  $b_i$ 。

接下来  $m$  行每行包含 3 个整数  $x, y, w$ ,含义如题所示。

### 【输出格式】

输出到文件 `transfer.out` 中。

一行包含  $K$  个整数,第  $i$  个整数表示  $k = i$  时的答案。

### 【样例 1 输入】

```
1 5 0 5
2 1 2 3 4 5
3 1 2 3 4 5
```

### 【样例 1 输出】

```
1 14 28 42 56 58
```



见选手目录下的 *transfer/transfer1.in* 与 *transfer/transfer1.ans*。

**【样例 2】**

见选手目录下的 *transfer/transfer2.in* 与 *transfer/transfer2.ans*，该组样例满足测试点 1 的条件。

**【样例 3】**

见选手目录下的 *transfer/transfer3.in* 与 *transfer/transfer3.ans*，该组样例满足测试点 5 的条件。

**【样例 4】**

见选手目录下的 *transfer/transfer4.in* 与 *transfer/transfer4.ans*，该组样例满足测试点 8 的条件。

**【样例 5】**

见选手目录下的 *transfer/transfer5.in* 与 *transfer/transfer5.ans*，该组样例满足测试点 14 的条件。

**【样例 6】**

见选手目录下的 *transfer/transfer6.in* 与 *transfer/transfer6.ans*，该组样例满足测试点 19 的条件。

**【样例 7】**

见选手目录下的 *transfer/transfer7.in* 与 *transfer/transfer7.ans*，该组样例满足测试点 22 的条件。

**【数据范围】**

对于 100% 的数据， $1 \leq n \leq 10^5$ ， $0 \leq m \leq 5 \times 10^5$ ， $1 \leq K \leq \min(300, n)$ ， $0 \leq a_i, b_i < 2^{12}$ ， $0 \leq w \leq 10^5$ 。

保证不存在两条信息其  $x, y$  完全相同。

测试点编号	$n$	$m$	$K$
1 ~ 2	$\leq 50$	$\leq 2500$	$\leq 50$
3 ~ 4	$\leq 300$	$\leq 10^4$	$\leq 300$
5 ~ 7	$\leq 10^5$	$\leq 5 \times 10^5$	$= 1$
8 ~ 10	$\leq 10^5$	$\leq 5 \times 10^5$	$\leq 5$
11 ~ 13	$\leq 5000$	$\leq 10^5$	$\leq 20$
14 ~ 16	$\leq 3 \times 10^4$	$\leq 2 \times 10^5$	$\leq 100$
17 ~ 18	$\leq 5 \times 10^4$	$\leq 3 \times 10^5$	$\leq 200$
19 ~ 21	$\leq 7 \times 10^4$	$\leq 5 \times 10^5$	$\leq 250$
22 ~ 25	$\leq 10^5$	$\leq 5 \times 10^5$	$\leq 300$