

# 省选 2024 模拟赛 Day 2

时间：2024 年 2 月

题目名称	游戏	摸牌	轻涟
题目类型	传统题	传统题	传统题
目录	game	cards	vaguelette
可执行文件名	game	cards	vaguelette
输入文件名	game.in	cards.in	vaguelette.in
输出文件名	game.out	cards.out	vaguelette.out
每个测试点时限	1 秒	1 秒	3 秒
内存限制	1024 MB	1024 MB	1024 MB
子任务数目	10	5	15
测试点是否等分	是	否	否

提交源程序文件名

对于 C++ 语言	game.cpp	cards.cpp	vaguelette.cpp
-----------	----------	-----------	----------------

编译选项

对于 C++ 语言	-O2 -std=c++14
-----------	----------------

注意事项（请仔细阅读）

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C/C++ 中函数 `main()` 的返回值类型必须是 `int`，程序正常结束时的返回值必须是 0。
3. 提交的程序代码文件的放置位置请参考各省的具体要求。
4. 因违反以上三点而出现的错误或问题，申诉时一律不予受理。
5. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
6. 选手提交的程序源文件必须不大于 100KB。
7. 程序可使用的栈空间内存限制与题目的内存限制一致。
8. 只提供 Linux 格式附加样例文件。
9. 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以此为准。

# 游戏 (game)

## 【题目描述】

给定一个长度为  $n$  的 01 序列  $a_1 \dots a_n$ 。定义一次 01 序列  $a$  的操作为：

- 设当前序列长度为  $m$ ，选择一个位置  $i$  满足  $a_i = 1$ ，然后删去  $a_{m-i+1}$ 。

定义多个 01 序列组成的游戏为：

- 先后手交替操作，每次可以任选一个可以操作的序列操作任意次，不能操作者输。

现在对  $a$  进行  $m$  次修改或询问，每次给定参数  $opt$  表示当前为修改还是询问：

- 若  $opt = 1$ ，则再给定两个数  $x, y$ ，表示将区间  $[x, y]$  中每个数取反。
- 若  $opt = 2$ ，则再给定一个数  $k$  和  $k$  个区间  $[l_1, r_1], [l_2, r_2], \dots, [l_k, r_k]$ ，拿出这些区间组成的  $k$  个序列进行游戏，询问先手是否能获胜，若能则输出 Yes，否则输出 No。

注意：游戏不影响原序列，每个区间拿出来之后就独立了，不影响其他区间。

## 【输入格式】

从文件 `game.in` 中读入数据。

第一行两个正整数  $n, q$ 。

第二行  $n$  个整数，表示序列  $a$ 。

接下来  $q$  行，每行先读入一个数  $opt$ ：

- 若  $opt = 1$ ，则同一行再读入两个数  $x, y$ ，表示一次修改。
- 若  $opt = 2$ ，则先在同一行读入一个正整数  $k$ ，接下来  $k$  行，每行两个正整数  $l_i, r_i$ ，意义如题面所示。

## 【输出格式】

输出到文件 `game.out` 中。

对于每个询问，输出一行表示答案。

## 【样例 1 输入】

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

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

【样例 1 输出】

```
1 Yes
2 Yes
```

【样例 2 输入】

```
1 5 5
2 0 1 1 1 1
3 1 1 5
4 2 2
5 4 5
6 4 4
7 1 3 5
8 1 4 4
9 2 2
10 1 2
11 3 4
```

【样例 2 输出】

```
1 No
2 No
```

【样例 3】

见选手目录下的 *game/game3.in* 与 *game/game3.ans*。

【数据范围】

对于 100% 的数据， $1 \leq n \leq 10^5, 1 \leq \sum k \leq 10^5, a_i \in \{0, 1\}$ 。

测试点编号	$n, m \leq$	特殊性质
1	5	$k \leq 2$
2, 3	$10^3$	$k \leq 10$
4	$10^5$	$k = 1$
5	$10^5$	$l_i = r_i$
6	$10^5$	所有 2 操作在 1 操作后
7, 8, 9, 10	$10^5$	

## 摸牌 (cards)

### 【题目描述】

小青蛙有写着  $1 \sim n$  的卡牌各  $m$  张。初始他每张卡牌手中各有  $a_i$  张，剩余的均在牌山中。保证  $\sum_{i=1}^n a_i = m$ 。

小青蛙希望他所有的卡牌写有同样的数字，然而他不会概率论，所以只会随机挑选一张打出去。

与此同时，牌山也会随机选择一张牌山中的卡牌发给他。你可以理解为，小青蛙从手中的  $m$  张牌随机选择一张，而牌山从剩下的  $(n-1)m$  张牌中随机选择一张与小青蛙选择的这张牌交换。

小青蛙不想等太长时间，于是他想知道自己期望打出去多少张牌后他手中的所有牌写有同样的数字，答案对 998244353 取模。

另外，小青蛙想多玩几局卡牌游戏，但是都使用同一副牌。也就是说，本题有  $t$  组数据，数据中唯一不同的是初始手中卡牌数量，即  $\{a\}$ 。

### 【输入格式】

从文件 `cards.in` 中读入数据。  
第一行三个正整数  $n, m, t$ ，意义如题面所示。  
接下来  $t$  行，每行  $n$  个正整数  $a_1, a_2, \dots, a_n$ 。保证  $\sum_{i=1}^n a_i = m$ 。

### 【输出格式】

输出到文件 `cards.out` 中。  
共  $t$  行，每行一个整数，表示期望打出的牌数对 998244353 取模的值。

### 【样例 1 输入】

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

【样例 1 输出】

```
1 0
2 9
```

【样例 1 解释】

$a_1 = 0, a_2 = 3$  时，初始状态就写有同样数字，故答案为 0。

$a_1 = 1, a_2 = 2$  时，当且仅当打出 1，摸进 2 时才会使得所有牌写有同样数字，否则将变为等价状态。其发生概率为  $\frac{1}{9}$ ，故答案为 9。

【样例 2 输入】

```
1 6 7 2
2 1 2 0 3 0 1
3 1 1 1 2 2 0
```

【样例 2 输出】

```
1 825297686
2 559099197
```

【数据范围】

对于 100% 的数据,  $1 \leq m \leq 10^5, 1 \leq n \leq 5 \times 10^3, 1 \leq t \leq 200, 0 \leq a_i \leq m, \sum_{i=1}^n a_i = m$ 。

子任务编号	$n \leq$	$m \leq$	子任务分值
1	5	5	12
2	17	17	21
3	2	$10^5$	17
4	400	$10^5$	26
5	$5 \times 10^3$	$10^5$	24

## 轻涟 (vaguelette)

### 【题目描述】

小青蛙在学习轻涟剖分。

对于一棵  $n$  个节点，边有边权的树  $T$  和长度为  $m$  ( $m$  为偶数)，值域在  $[1, n]$  内的序列  $a$ ，定义它的贡献  $F(T, a)$  表示将  $a$  中元素两两配对后在树上形成的  $\frac{m}{2}$  条路径的长度和的最小值。注意如果有重复元素需要配对多次。

同时，定义它的轻呱值  $G(T, a)$  表示对于  $a$  的每个长为偶数的子序列  $a'$ ， $F(T, a')$  的和。

再定义轻涟值  $H(T, m)$  表示对于任意合法的，长度为  $m$  的序列  $p$ ， $G(T, p)$  的和。

小青蛙使用轻涟剖分轻松的解决了这个问题，于是他的好朋友小青呱决定加强一下。

**问题一：**给定  $n$  个节点的树  $T$  和偶数  $m$ ，请对于其中的所有连通块  $T'$ ，求  $H(T', m)$  的和。

小青蛙再次使用轻涟剖分轻松的解决了这个问题。但这之后小青蛙便忙着去玩卡牌游戏了，等到回来之后发现自己只记得前  $k$  个点跟它们父亲之间的边权，忘记了这些点的父亲。不过他还记得一件事情，那就是每个非根结点的父亲编号都小于它自身的编号。

**问题二：**小青蛙想知道，在仅保留前  $k$  个点的情况下，所有合法的树中，问题一的答案之和。

小青蛙的脑容量不算很大（从他健忘就可以看出来），所以你只要求出这两问的答案对  $10^9 + 7$  取模的值。

### 【输入格式】

从文件 `vaguelette.in` 中读入数据。

第一行三个整数  $n, m, k$ ，意义如题面所示。

第二行  $n - 1$  个整数，其中第  $i$  个数  $fa_{i+1}$  表示  $i + 1$  的父亲。

第三行  $n - 1$  个整数，其中第  $i$  个数  $val_{i+1}$  表示  $i + 1$  的与它父亲之间的边权。

### 【输出格式】

输出到文件 `vaguelette.out` 中。

一行两个数，分别表示两问的答案。

第一问正确会获得 60% 的分数，第二问正确会获得 40% 的分数。如果你不会某一问，请随便输出一个数以保证格式正确。每个子任务的评分取该子任务中每个测试点的得分最小值。

【样例 1 输入】

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

【样例 1 输出】

```
1 4944 28976
```

【样例 2 输入】

```
1 7 7 7
2 1 1 2 2 3 3
3 5 4 3 2 1 2
```

【样例 2 输出】

```
1 42528697 1655828
```

【样例 3 ~ 17】

见选手目录下的 *vaguelette/vaguelette3 ~ 17.in* 与 *vaguelette/vaguelette3 ~ 17.ans*。

对于  $i \in [1, 15]$ ，第  $i + 2$  个样例满足子任务  $i$  的限制。

【数据范围】

对于 100% 的数据， $1 \leq n \leq 7500, 2 \leq m \leq 10^9, 2 \mid m, 1 \leq k \leq 500, 1 \leq fa_i < i, 1 \leq val_i \leq 10^9$ 。

保证除前两个样例外，均有  $1 \leq k \leq \lceil \frac{n}{15} \rceil$ 。



子任务编号	$n \leq$	$m \leq$	特殊性质	子任务分值
1	4	2		10
2	20	20		5
3	50	50		5
4	100	100		5
5	200	200		5
6	300	300		5
7	5000	$10^9$	$fa_i = i - 1$	5
8	5000	$10^9$	$fa_i = 1$	10
9	5000	2		5
10	1250	$10^9$		5
12	2500	$10^9$		10
12	3750	$10^9$		5
13	5000	$10^9$		10
14	6250	$10^9$		5
15	7500	$10^9$		10