D1t4b. 简单难题练习题

时间限制：1.0s　**内存限制**：256.0MB
输入文件名：**easyhard.in**　输出文件名：**easyhard.out**

附加文件：
samples.rar (462B)

Description

「朝露」

蓬莱山辉夜向你打出了最后一张符卡

古老的力量开始涌动，静谧的夜幕不断翻滚

五色的琉璃弹幕交织在一起

似是宣告那黎明的到来——正如其名一般

在这无尽的永夜返中，你只看到了辉夜留给你的最终谜题：

盒子里有n个球，球的颜色有黑白两种，但是不知道初始状态

一共进行m次操作，每次会从盒子中取出一个球，然后放入黑白各一个球，然后再取出一个球

取出的$2m$个球会形成一个序列，你需要对所有初始状态计算本质不同的序列对p取模的结果

两个序列不同当且仅当存在一个位置，其所代表的球的颜色在两个序列中不同

这里的初始状态指的是盒子中两种颜色的球的个数

Input

一行三个数n, m, p

Output

一行一个数表示答案

Sample Input

```
1 2 114514
```

Sample Output

```
8
```

Data Constraint

对于100的数据，$1 \leq n, m \leq 3 * 10^3, 1 \leq p \leq 10^9$

测试点	n	m
1	≤ 10	≤ 10
2	≤ 30	≤ 30
$3-4$	≤ 60	≤ 60
$5-6$	≤ 300	≤ 300
$7-10$	≤ 3000	≤ 3000

Hint

样例解释：

令黑为1，白为0

则8种不同的序列分别为$1010, 1011, 1100, 1101, 0010, 0011, 0100, 0101$

D5vto. 往事成风

时间限制：1.0s　**内存限制**：512.0MB
输入文件名：**imperishable.in**　输出文件名：**imperishable.out**

附加文件：

imperishable1.ans (10B) | imperishable1.in (28B) | imperishable2.ans (25.65KB) | imperishable2.in (40.21KB)

题目描述

> 几年之后全世界会刮起一场大风
> 我们会因这场风而各奔东西
> 但心却紧紧相系

定义一个可重集 S 的众数 $f(S)$ 为集合中出现次数最多且编号最小的数字。例如，$f(1,2,2,3)=2$，$f(2,2,3,3)=2$。

定义一个可重集 S 的价值为 $2^{f(S)}$。

现在你有一个长度为 n 的序列 $a_1,a_2,\cdots,a_n$，其中数字 a_x 表示可重集 S 中正整数 x 出现了 a_x 次。你需要将这个可重集 S 不重、不漏地划分成若干个可重集，使得这些可重集的价值之和最小。

你只需要输出这个数值在模 998244353 意义下的结果。

接下来，会有 q 次修改，每次修改会将 a_x 改为 y。每次修改后你都要重新输出上面所描述的数值。

输入格式

从文件 imperishable.in 中读入数据。

第一行一个正整数 Num，表示测试点编号。你可能不需要这个数字。

接下来一行一个正整数 n，表示序列的长度。

接下来一行 n 个正整数，分别表示 $a_1,a_2,\ldots,a_n$。

接下来一个正整数 q，表示修改次数。

接下来 q 行，每行两个正整数 x,y，表示将 a_x 改为 y。

输出格式

输出到文件 imperishable.out 中。

答案对 998244353 取模。

在修改之前，先输出一个数值表示答案。

每次修改之后，都要输出一个数值表示答案。

样例输入 1

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

样例输出 1

```
2
10
6
```

样例 1 解释

刚开始，可重集为 $1,1,1,1,2,3,3,4,4,4$，可以划分为可重集 $1,1,1,1,2,3,3,4,4,4$，答案为 $2^1=2$。

第一次修改后，可重集变为 $1,2,3,3,4,4,4$，可将其划分为 $1,4$ 和 $2,3,3,4,4$，答案为 $2^1+2^3=10$。

第二次修改后，可重集变为 $1,1,2,3,3,4,4,4$，可将其划分为 $1,1,3,4,4$ 和 $2,3,4$ ，答案为 $2^1+2^2=6$。

样例 2

见下发数据下的 `imperishable2.in` 和 `imperishable2.ans` 。

数据范围

对于所有数据，满足 $1\le n\le 2\times 10^5,0\le q\le 2\times 10^5,1\le a_i,y\le 2\times 10^5,1\le x\le n$。

数据有一定梯度。

测试点编号	n	q	$max\{a_i,y\}$	特殊性质
1	≤ 1	≤ 0	≤ 1	无
$2\sim 5$	≤ 5	≤ 2	≤ 10	$\sum a_i\le 10$
6	≤ 15	≤ 2	≤ 10	无
$7\sim 11$	≤ 60	≤ 50	≤ 10	无
$12\sim 14$	≤ 3000	≤ 3000	≤ 4	无
15	≤ 3000	≤ 3000	$\le 2\times 10^5$	无
16	≤ 5000	≤ 5000	$\le 2\times 10^5$	无
17	≤ 10000	≤ 10000	$\le 2\times 10^5$	无
$18\sim 20$	$\le 5\times 10^4$	$\le 10^5$	$\le 2\times 10^5$	任意时刻，a_i 互不相同
$21\sim 22$	$\le 5\times 10^4$	$\le 5\times 10^4$	$\le 2\times 10^5$	无
$23\sim 25$	$\le 2\times 10^5$	$\le 2\times 10^5$	$\le 2\times 10^5$	无

Dhi1e. Ginger 的无向无环联通图

时间限制：1.5s　**内存限制**：2.0GB　**代码提交间隔**：3分钟(现在可以提交)
输入文件名：**treeq.in**　输出文件名：**treeq.out**
试题来源：**姜锐漳**

附加文件：
ex_treeq.ans (6B)　ex_treeq.in (178.9MB)

题目描述

有一棵 n 个节点的有根树，一条树边可以记为 (F_i, w_i) ，表示连接点 F_i 和点 i 权为 w_i。

对于边 i ， $a_{i,j}$ 的定义如下：令 $c_{j,i}$ 为点 j 与其他 $n-1$ 个点之间的简单路径中经过边 i 的路径数，那么 $a_{i,j}=c_{j,i}w_i$ 。

一条边 i 在点 j 是优的，当且仅当不存在有另一条边 k 使得 $a_{k,j}>a_{i,j}$ （可能有多条边在同一个点是优的）。

令 $f(i)$ 表示边 i 在多少个点上是优的，现在要求 $f(i)$。

输入格式

第一行，一个正整数 n。

接下来 $n-1$ 行，第 i 行两个正整数 F_{i+1}, w_{i+1} 表示一条边的信息。

输出格式

为减少输出量，共输出一行，一个整数，表示所有 $f(i)$ 的异或和。

样例输入

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

样例输出

```
2
```

数据范围

数据点编号	数据范围	特殊性质
$1,2$	$n\leq 5\ 000$	
$3,4$	$n\leq 10^6$	$w_i=1$
$5,6$	$n\leq 10^6$	$F_i=i-1$
$7,8$	$n\leq 10^6$	$F_i=1$
$9\sim 15$	$n\leq 10^6$	
$16\sim 20$	$n\leq 10^7$	

对于所有数据， $n\leq 10^7, w_i\leq 10^9, F_i<i$ 。

请选手注意常数。

样例解释

各条边的 f 分别为 $3,1,1,1$ 。

Dp9jc. 走廊

时间限制：1.5s　**内存限制**：1.0GB
输入文件名：**corridor.in**　输出文件名：**corridor.out**

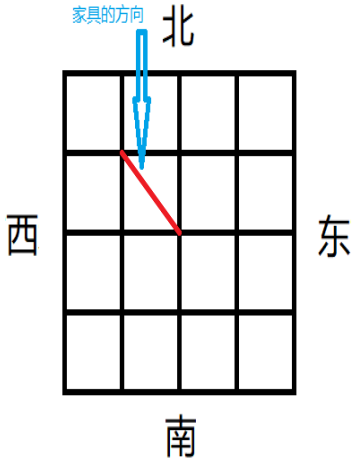
附加文件：

corridor1.ans (20B)　corridor1.in (60B)　corridor2.ans (48.15KB)　corridor2.in (118.9KB)

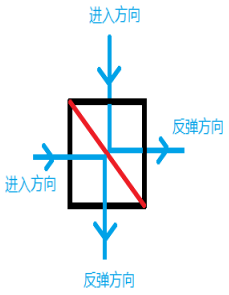
题目描述

GreenDuck 买到了最新的扫地机器人——RobotDuck。为了更高效地清扫房间，GreenDuck 决定分析一下房间的布局和 RobotDuck 的清扫机制。

GreenDuck 的房间可以看成是 n 行 m 列的网格，每个格子中要么是空的，要么放了一件家具。同时他惊奇地发现，每个家具都很薄(可以看成是一条线段)，两个端点分别占据了一个格子的西北角和东南角！



RobotDuck 在清扫时会采取这样一种机制：刚开始，可以给它设定一个方向(向南或向东)，接着它将一直沿着直线行进。如果碰到一个家具，那么会反弹。具体方式如下图所示。



GreenDuck 一开始得知，在自己的房间里有 k 件家具，第 i 件家具在第 x_i 行，y_i 列。接下来，他会进行 Q 次测试。

第一种测试，是在一个没有家具的格子上放上一件家具(方向保持一致)。这种测试后，他不会拿走任何家具。

第二种测试，是将 RobotDuck 贴着墙壁然后释放。具体来说，如果 RobotDuck 放在北面(对应了图中矩形的上边界)，那么它会从第一行某个格子的上边界的中心出发，面向南面行进。如果 RobotDuck 放在西面(对应了图中矩形的左边界)，那么它会从第一列某个格子的左边界的中心出发，面向东面行进。在行进过程中，它遵守清扫的机制。

GreenDuck 想知道，在每次第二种测试时，RobotDuck 在反弹恰好 q 次后会在哪个格子里。请你告诉他。

输入格式

第一行四个数字，$type, n, m, k$，分别表示数据类型(你可能不需要)，行数，列数，一开始有的家具件数。

接下来 k 行，每行两个整数 x_i, y_i，分别表示这件家具的行数和列数。

接下来一行一个整数 Q，表示测试的次数。

接下来 Q 行，首先输入一个数字 w。

若 w 为 1，则会有两个数字 x, y，分别表示新添的家具的行数和列数。

若 w 为 2，则会有三个数字 x, y, q。若 $x = 0$，表示 RobotDuck 从第一行第 y 列格子的上边界的中心面向南行进。否则 $y = 0$，表示 RobotDuck 从第一列第 x 行格子的左边界的中心面向东行进。q 表示反弹的次数。

输出格式

对于每个 $w=2$ 的操作，输出一行两个数字 x 和 y 分别表示所在格子的行数和列数。

特别地，若 RobotDuck 在第 q 次反弹之前就碰到了墙壁，若墙壁是第 n 行格子的下边界，输出"$n+1$ 列数"，否则输出"行数 $m+1$"。

样例输入 1

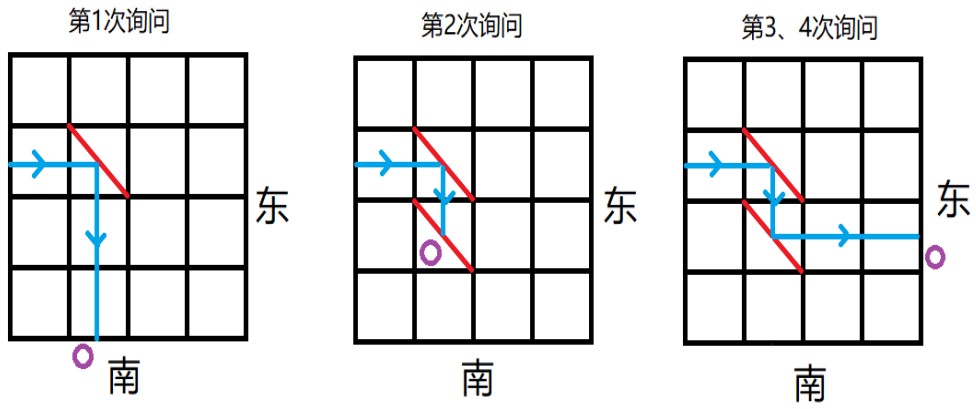
```
0 4 4 1
2 2
5
2 2 0 3
1 3 2
2 2 0 2
2 2 0 3
2 2 0 4
```

样例输出 1

```
5 2
3 2
3 5
3 5
```

样例 1 解释

下图中紫色的圆圈表示最终所到达的格子。



样例 2

见下发文件下的 `corridor2.in` 和 `corridor2.ans` 。

数据范围

对于所有数据，家具的坐标不会重复，每次第二种操作要么 $x=0$ 要么 $y=0$。

$1 \le q \le 200000$。

测试点编号	$type$	n, m	$k+Q$	特殊性质
$1 \sim 5$	$=0$	≤ 100	≤ 2000	无
$6 \sim 11$	$=1$	≤ 20000	≤ 150000	询问中不会出现1号操作
$12 \sim 17$	$=2$	≤ 50000	≤ 150000	反弹次数之和不超过 10^7 (不是 q 之和)
$18 \sim 25$	$=3$	≤ 50000	≤ 150000	无