

# 2025 山东省队第三轮集训 历城站 Contest 3

LeafSeek

2025 年 5 月 28 日

题目名称	整体论	还原论	无论
题目类型	传统题	传统题	传统题
目录	alpha	sigma	omega
可执行文件名	alpha	sigma	omega
输入文件名	alpha.in	sigma.in	omega.in
输出文件名	alpha.out	sigma.out	omega.out
测试点时间限制	2 秒	2 秒	2 秒
测试点空间限制	512 MB	512 MB	512 MB
题目总分值	100	100	100
子任务个数	6	8	10
子任务是否等分	否	否	是

## A. 整体论

### 题目描述

小  $\alpha$  给了你一个长度为  $n$  的序列  $a_1, a_2, \dots, a_n$ 。小  $\alpha$  有  $q$  个操作，每个操作为：

1. 给定  $c$ ，令  $a_i \leftarrow \min(a_i, c)$ ，对每个  $1 \leq i \leq n$ 。
2. 给定  $l, r, c$ ，令  $a_i \leftarrow \max(a_i, c)$ ，对每个  $l \leq i \leq r$ 。

一共有  $q!$  种不同的操作顺序，对每种顺序，将它们依次操作到初始序列  $a_1, a_2, \dots, a_n$  后都能得到一个最终序列。问有多少种可能的不同最终序列，对 998244353 取模。

### 输入格式

从文件 `alpha.in` 中读入数据。

第一行三个正整数  $n, m, k$ ，表示序列的长度，第一种操作的个数和第二种操作的个数。

第二行  $n$  个整数，表示初始序列  $a_1, a_2, \dots, a_n$ 。

第三行  $m$  个整数，表示所有第一种操作的参数  $c$ 。

接下来  $k$  行，每行三个整数  $l, r, c$ ，表示一个操作。

### 输出格式

输出到文件 `alpha.out` 中。

一行一个正整数，表示最终序列个数对 998244353 取模的结果。

### 样例 0

#### 样例 0 输入

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

#### 样例 0 输出

```
6
```

### 样例 1 ~ 10

#### 样例 1 ~ 10 输入

见选手目录下的 `alpha/ex_alpha1-10.in`。

#### 样例 1 ~ 10 输出

见选手目录下的 `alpha/ex_alpha1-10.out`。

### 数据范围与提示

对于所有数据，保证  $1 \leq n, m, k \leq 150$ 。

子任务编号	子任务分值	$n \leq$	$m \leq$	$k \leq$	特殊性质
1	10	20	3	7	
2	20		1		
3	20		5		
4	10				保证 $l = r$
5	20	30	30	30	
6	20				

## B. 还原论

### 题目描述

小  $\sigma$  给了你一个长度为  $n$  的序列  $a_1, a_2, \dots, a_n$ 。小  $\sigma$  需要你执行共  $q$  次下面两种操作：

- 1 1  $r$ ：将  $a[l, r]$  替换为它的异或差分。形式化地说，对于每个  $l < i \leq r$ ，令  $b_i = a_i \text{ xor } a_{i-1}$ ，然后对于每个  $l < i \leq r$ ，将  $a_i$  替换为  $b_i$ 。
- 2 2  $\text{pos}$ ：查询  $a_{\text{pos}}$  的值。

所有操作执行完后，你还需要回答最终的  $a$  序列。

### 输入格式

从文件 `sigma.in` 中读入数据。

第一行包含一个整数  $T$ ，表示该数据满足第  $T$  个子任务的限制。

第二行包含两个整数  $n, q$ ，分别表示序列的长度和操作的个数。

第三行包含  $n$  个整数  $a_1, a_2, \dots, a_n$ 。

接下来  $q$  行，每行若干个数，表示一个操作。若操作为第一种操作，则此行包含三个数 1 1  $r$ 。若操作为第二种操作，则此行包含两个数 2  $\text{pos}$ 。

### 输出格式

输出到文件 `sigma.out` 中。

设共有  $q_2$  个第二种操作，则输出共包含  $q_2 + n$  行。

前  $q_2$  行，每行输出一个整数，表示该操作的答案。

接下来  $n$  行，每行输出一个整数，表示最终的  $a$  序列。

### 样例 0

#### 样例 0 输入

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

#### 样例 0 输出

```
9
4
12
1
0
5
4
12
0
```

#### 样例 0 解释

初始时  $a = [1, 1, 5, 1, 9, 4]$ 。

第一个操作要求将  $a_{[2,5]}$  替换为它的异或差分， $a_{[2,5]}$  为  $[1, 5, 1, 9]$ ，它的异或差分为  $[1, 4, 4, 8]$ ，故操作执行完后， $a$  序列变为  $[1, 1, 4, 4, 8, 4]$ 。

第二个操作查询  $a_5$  的值，此时  $a_5 = 9$ ，故输出 9。

第三个操作要求将  $a_{[3,6]}$  替换为它的异或差分,  $a_{[3,6]}$  为  $[4, 4, 8, 4]$ , 它的异或差分为  $[4, 0, 12, 12]$ , 故操作执行完后,  $a$  序列变为  $[1, 1, 4, 0, 12, 12]$ 。

第四个操作查询  $a_4$  的值, 此时  $a_4 = 4$ , 故输出 4。

第五个操作要求将  $a_{[1,6]}$  替换为它的异或差分,  $a_{[1,6]}$  为  $[1, 1, 4, 0, 12, 12]$ , 它的异或差分为  $[1, 0, 5, 4, 12, 0]$ , 故操作执行完后,  $a$  序列变为  $[1, 0, 5, 4, 12, 0]$ 。

最终的  $a$  序列为  $[1, 0, 5, 4, 12, 0]$ 。

## 样例 1 ~ 8

### 样例 1 ~ 8 输入

见选手目录下的 `sigma/ex_signal-8.in`。

### 样例 1 ~ 8 输出

见选手目录下的 `sigma/ex_signal-8.out`。

## 数据范围与提示

对于所有数据, 保证  $1 \leq n \leq 2.5 \times 10^5$ ,  $1 \leq q \leq 10^5$ ,  $0 \leq a_i < 2^{30}$ ,  $1 \leq l \leq r \leq n$ ,  $1 \leq \text{pos} \leq n$ 。

子任务编号	子任务分值	$n \leq$	$q \leq$	特殊性质
1	10	$2 \times 10^3$	$2 \times 10^3$	无
2	10	$2.5 \times 10^5$	$10^5$	A
3	10	$2.5 \times 10^5$	$10^5$	B
4	10	$2.5 \times 10^5$	$10^5$	CD
5	10	$2.5 \times 10^5$	$10^5$	DE
6	10	$2.5 \times 10^5$	$10^5$	D
7	10	$2.5 \times 10^5$	$10^5$	E
8	30	$2.5 \times 10^5$	$10^5$	无

特殊性质 A:  $\forall i \geq 2, a_i = 0$ 。

特殊性质 B:  $0 \leq a_i \leq 1$ 。

特殊性质 C: 记序列  $a$  中非零位置个数为  $c$ , 则  $c \leq 100$ 。

特殊性质 D: 操作 1 满足  $l = 1, r = n$ 。

特殊性质 E: 没有操作 2。

## C. 无论

### 题目描述

小  $\omega$  种了一排一共  $n$  株草，第  $i$  株草有一个原始高度  $a_i$ ，还有一个目标高度  $b_i$ 。

我们认为高度  $> 0$  的草是存活的，而高度  $= 0$  的草已经死亡。

小  $\omega$  想要修剪这些草，有两种修剪操作：

- 选定一个区间  $[l, r]$ ，将  $[l, r]$  内所有仍然存活的草的高度都增加 1；
- 选定一个区间  $[l, r]$ ，将  $[l, r]$  内所有仍然存活的草的高度都削减 1；

如果一株草在经过这个操作后高度变为了 0，那么就被视为已经死亡，这株草不再受后续的操作影响。

你需要求出让所有草的高度从  $a_i$  变成  $b_i$  所需要的修剪操作的最小个数。

### 输入格式

从文件 `omega.in` 中读入数据。

一个测试点中包含多组测试数据。

输入的第一行包含一个整数  $T$ ，表示数据组数。

对于每组数据，第一行一个整数  $n$ 。

接下来一行，包含  $n$  个整数，描述  $a_1, a_2, \dots, a_n$ 。

接下来一行，包含  $n$  个整数，描述  $b_1, b_2, \dots, b_n$ 。

### 输出格式

输出到文件 `omega.out` 中。

对于每组数据，输出一行一个整数，表示答案。若合法方案不存在，则输出  $-1$ 。

### 样例 0

#### 样例 0 输入

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

#### 样例 0 输出

```
3
18
4
```

### 样例 1 ~ 10

#### 样例 1 ~ 10 输入

见选手目录下的 `omega/ex_omega1-10.in`。

#### 样例 1 ~ 10 输出

见选手目录下的 `omega/ex_omega1-10.out`。

## 数据范围与提示

对于所有数据，保证  $1 \leq n \leq 10^6$ ， $1 \leq \sum n \leq 3 \times 10^6$ ， $0 \leq a_i, b_i \leq 10^9$ 。  
以下展示每个子任务各自的数据范围以及特殊限制：

子任务编号	子任务分值	$T \leq$	$n \leq$	$a_i, b_i \leq$	特殊限制
1	10		3		
2	10		4		
3	10		5		
4	10	10	100	100	
5	10	10	500	500	
6	10	10	3000	500	
7	10	10	$10^5$	500	
8	10				$b_i > 0$
9	10				$a_i, b_i$ 单调不减
10	10				