

# CSP-S 模拟赛 day6

时间：2022 年 10 月 6 日 ??:?? ~ ??:??

题目名称	前 k 大	石头剪刀布	吉司机线段树	排列
题目类型	传统型	传统型	传统型	传统型
输入文件名	kth.in	rps.in	sgtbeats.in	permutation.in
输出文件名	kth.out	rps.out	sgtbeats.out	permutation.out
每个测试点时限	1.5 秒	1.0 秒	1.5 秒	1.5 秒
内存限制	512 MB	512 MB	512 MB	512 MB
测试点数目	10	10	10	10
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	kth.cpp	rps.cpp	sgtbeats.cpp	permutation.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. 程序可使用的栈空间内存限制与题目的内存限制一致。
6. 题目不一定按照难度顺序排序，请注意掌握时间。

## 前 k 大 (kth)

### 【题目描述】

小 T 刚刚学会二进制。

对于正整数  $x$ ，定义  $f(x)$  为  $x$  的二进制表示中，从最低位开始的连续的零的个数，例如  $f(2) = 1, f(5) = 0, f(40) = 3$ 。

定义序列  $a$  为  $a_n = n + c \times f(n)$ ，其中  $c$  是一个给定的非负整数。

你需要求出  $a_l, a_{l+1}, \dots, a_r$  中前  $k$  大元素之和，换句话说，求出将  $a_l, a_{l+1}, \dots, a_r$  从大到小排序后前  $k$  个数的和。

### 【输入格式】

从文件 **kth.in** 中读入数据。

本题输入文件包含多组数据。

第一行一个整数  $T$ ，表示数据组数。

接下来  $T$  行，每行四个整数  $l, r, k, c$ ，含义见题目描述。

### 【输出格式】

输出到文件 **kth.out** 中。

输出  $T$  行，每行一个整数，表示这组数据的答案。

### 【样例 1 输入】

```
2
5 7 2 2
6 11 5 1
```

### 【样例 1 输出】

```
15
49
```

### 【样例 1 解释】

对于第一组数据， $[a_l, a_{l+1}, \dots, a_r] = [5, 8, 7]$ ，答案为  $8 + 7 = 15$ 。

对于第二组数据， $[a_l, a_{l+1}, \dots, a_r] = [7, 7, 11, 9, 11, 11]$ ，答案为  $11 + 11 + 11 + 9 + 7 = 49$ 。

### 【样例 2】

见选手目录下的 **kth/kth2.in** 与 **kth/kth2.ans**。

**【样例 3】**

见选手目录下的 *kth/kth3.in* 与 *kth/kth3.ans*。

**【数据范围与提示】**

对于 30% 的数据,  $T \leq 10$ ,  $r - l + 1 \leq 10^5$ 。

对于 50% 的数据,  $T \leq 10$ ,  $k \leq 10^5$ 。

对于另外 10% 的数据,  $k = r - l + 1$ 。

对于 100% 的数据,  $1 \leq T \leq 10^5$ ,  $1 \leq l \leq r \leq 10^9$ ,  $1 \leq c \leq 10^9$ ,  $1 \leq k \leq r - l + 1$ 。

## 石头剪刀布 (rps)

### 【题目描述】

有  $n$  个人排成一排，在参加石头剪刀布大赛，每个人从头到尾只会使用一种手势。

每个时刻，都有且仅有某两个相邻的人进行石头剪刀布，若平局则无事发生，否则输的人出局，剩下的人依照原来的相对顺序排成一排。

当无法再有人出局的时候，大赛结束，仍未出局的人获胜，可以发现，获胜的所有人都使用了同一种手势。

现在大赛尚未开始，你知道其中一些人使用的手势。具体来说，你有一个字符串  $s$ ， $s_i = \text{R/P/S}$  分别表示从左往右数第  $i$  个人使用的手势为石头/布/剪刀， $s_i = ?$  表示第  $i$  个人使用的手势未知。

你要求出，在获胜的人使用石头/布/剪刀的条件下，获胜的人数至多可能是多少。这里的“可能”指存在一组未知手势的人使用的手势，以及每个时刻是哪两个人进行石头剪刀布。

### 【输入格式】

从文件 `rps.in` 中读入数据。

一行一个长为  $n$  的字符串  $s$ ，表示目前知道的手势信息。

### 【输出格式】

输出到文件 `rps.out` 中。

共三行，每行一个整数，分别为  $ans_R, ans_P, ans_S$ ，表示在获胜的人使用石头/布/剪刀的条件下，获胜的人数至多可能是多少。

### 【样例 1 输入】

PPSR?

### 【样例 1 输出】

2  
3  
1

### 【样例 1 解释】

最大化石头的方案：PPSRR  $\rightarrow$  PSRR  $\rightarrow$  SRR  $\rightarrow$  RR。

最大化布的方案：PPSRP  $\rightarrow$  PPRP  $\rightarrow$  PPP。

最大化剪刀的方案：PPSRP  $\rightarrow$  PSRP  $\rightarrow$  SRP  $\rightarrow$  SP  $\rightarrow$  S。

**【样例 2】**

见选手目录下的 *rps/rps2.in* 与 *rps/rps2.ans*。

**【样例 3】**

见选手目录下的 *rps/rps3.in* 与 *rps/rps3.ans*。

**【样例 4】**

见选手目录下的 *rps/rps4.in* 与 *rps/rps4.ans*。

**【数据范围与提示】**

对于 20% 的数据， $n \leq 10$ ，保证  $s$  中没有 ?。

对于 50% 的数据， $n \leq 300$ 。

对于另外 20% 的数据，保证  $s$  中没有 ?。

对于 100% 的数据， $1 \leq n \leq 10^6$ ， $s_i \in \{R, P, S, ?\}$ 。

## 吉司机线段树 (sgtbeats)

### 【题目描述】

小 T 想出数据结构题。

有一天，他想到了下面这个题。

- 开始给定一个长为  $n$  的序列  $a_i$ 。
- 有  $m$  次修改，每次修改为：
  - 形如  $1\ x$ ，表示将  $a_x$  修改为  $0$ 。
  - 形如  $2\ v$ ，表示将  $1 \leq i \leq n$  的每个数  $a_i$  修改为  $\max\{a_i, v\}$ 。
- 在修改完成之后，求序列中所有数的和。

小 T 把这道题告诉了小 W，小 W 说直接闭着眼拍一个吉司机线段树不就做完了吗。

小 T 决定加强这个题。

- 有  $q$  次删除操作，每次会删除原先的某个修改，你需要在每次删除操作后求出原问题的答案。保证删除的操作互不相同。
- 注意，删除操作是永久的，也即一旦删除了某个操作，它就永远不会被加入回来。在处理原问题时，序列的初值不会变化，依旧按顺序执行修改，但跳过所有被删除的操作。

小 T 不会做加强后的题了，于是又把这道题告诉了小 W，小 W 说直接闭着眼拍一个可追溯化吉司机线段树不就做完了吗。

小 T 很迷茫，所以请你来帮忙解决这个题。

### 【输入格式】

从文件 `sgtbeats.in` 中读入数据。

第一行三个整数  $n, m, q$ ，分别表示序列长度，修改次数，删除操作次数。

第二行  $n$  个整数  $a_i$ ，表示序列  $a_i$ 。

接下来  $m$  行，第  $i$  行两个整数形如  $1\ x$  或  $2\ v$ ，表示第  $i$  次操作。

接下来  $q$  行，每行一个整数  $d$ ，表示删除第  $d$  次修改。

### 【输出格式】

输出到文件 `sgtbeats.out` 中。

输出  $q$  行，每行一个整数，表示答案。

### 【样例 1 输入】

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

2 4  
1 5  
3  
5  
2  
6

### 【样例 1 输出】

23  
19  
15  
18

### 【样例 1 解释】

每次删除修改后的序列分别为:  $[7, 4, 6, 6, 0]$ ,  $[7, 0, 6, 6, 0]$ ,  $[7, 0, 3, 5, 0]$ ,  $[7, 0, 3, 5, 4]$ 。

### 【样例 2】

见选手目录下的 *sgtbeats/sgtbeats2.in* 与 *sgtbeats/sgtbeats2.ans*。

### 【样例 3】

见选手目录下的 *sgtbeats/sgtbeats3.in* 与 *sgtbeats/sgtbeats3.ans*。

### 【数据范围与提示】

对于 10% 的数据,  $n, m, q \leq 300$ 。

对于 30% 的数据,  $n, m, q \leq 5000$ 。

对于另外 20% 的数据, 删除操作只会删除形如  $1 \times$  的修改。

对于另外 20% 的数据, 保证数据随机。具体来说, 保证  $a_i, x, v, d$  在数据范围内等概率随机。

对于 100% 的数据,  $1 \leq n, m \leq 3 \times 10^5$ ,  $1 \leq q \leq m$ ,  $1 \leq x \leq n$ ,  $1 \leq v, a_i \leq 10^9$ ,  $1 \leq d \leq m$ , 保证删除的操作互不相同。

## 排列 (permutation)

### 【题目描述】

定义一个序列  $b$  是好的，当且仅当选择若干个（可以为 0）个两两不交的区间，对每个区间翻转后得到的序列严格单增。例如，序列  $[6, 4, 2, 9, 10, 15, 13, 20]$  是好的，因为选择区间  $[1, 3], [6, 7]$  翻转后序列变为  $[2, 4, 6, 9, 10, 13, 15, 20]$ ，满足严格单增。特别地，空序列  $[\ ]$  是好的。

给定一个长为  $n$  的排列  $a$ ，对于每个  $k$ ，你需要求出有多少子序列是好的，且不包含  $a_k$ 。你只需要输出答案对 998244353 取模的结果。注意，子序列可以为空。

### 【输入格式】

从文件 *permutation.in* 中读入数据。

第一行一个整数  $n$  表示排列长度。

第二行  $n$  个整数  $a_i$ ，表示序列  $a$ 。

### 【输出格式】

输出到文件 *permutation.out* 中。

输出一行， $n$  个整数，其中第  $k$  个整数  $ans_k$  表示不含  $a_k$  的好的子序列的数量对 998244353 取模的结果。

### 【样例 1 输入】

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

### 【样例 1 输出】

```
14 13 12 13 13
```

### 【样例 1 解释】

不包含  $a_1 = 5$  的子序列有 16 个，其中不好的子序列有  $[1, 4, 2, 3], [4, 2, 3]$  两个。

不包含  $a_2 = 1$  的子序列有 16 个，其中不好的子序列有  $[5, 4, 2, 3], [4, 2, 3], [5, 2, 3]$  三个。

不包含  $a_3 = 4$  的子序列有 16 个，其中不好的子序列有  $[5, 1, 2, 3], [5, 2, 3], [5, 1, 2], [5, 1, 3]$  四个。

其余答案可以自行验证。

### 【样例 2】

见选手目录下的 *permutation/permutation2.in* 与 *permutation/permutation2.ans*。



**【样例 3】**

见选手目录下的 *permutation/permutation3.in* 与 *permutation/permutation3.ans*。

**【数据范围与提示】**

对于 20% 的数据,  $n \leq 18$ 。

对于 40% 的数据,  $n \leq 300$ 。

对于另外 20% 的数据, 保证对于所有  $1 \leq i \leq n$ ,  $|a_i - i| \leq 5$ 。

对于 100% 的数据,  $1 \leq n \leq 3000$ , 保证  $a_i$  是一个  $1 \sim n$  的排列。

提示: 要相信现代计算机的速度。