

全国青少年信息学奥林匹克竞赛

NOIP2024模拟

时间：7:40-12:00

题目名称	游客与区间	游客与染色	游客与波特	游客与串串
题目类型	传统型	传统型	传统型	传统型
目录	interval	color	bot	string
可执行文件名	interval	color	bot	string
输入文件名	interval.in	color.in	bot.in	string.in
输出文件名	interval.out	color.out	bot.out	string.out
每个测试点时限	1.0秒	1.0秒	3.0秒	2.0秒
内存限制	512 MB	512 MB	512MB	512 MB
子任务数目	4	5	8	6
测试点是否等分	是	是	是	是

提交源程序文件名

对于C++语言	interval.cpp	color.cpp	bot.cpp	string.cpp
---------	--------------	-----------	---------	------------

编译选项

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

注意事项与提醒（请选手务必仔细阅读）

- 文件名（程序名和输入输出文件名）必须使用英文小写。
- C++ 中主函数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。
- 提交的程序代码文件的放置位置请参照各省的具体要求。
- 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。
- 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）。
- 程序可使用的栈内存空间限制与题目的内存限制一致。
- 全国统一评测时采用的机器配置为：Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz，内存 32GB。上述时限以此配置为准。
- 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其为准。
- 终评测时所用的编译命令中不含编译选项之外的任何优化开关。

1.游客与区间 (interval)

【题目描述】

tourist 有 n 个区间 $[l_1, r_1], [l_2, r_2], \dots, [l_n, r_n]$ 。他想划分这些区间，我们称一个划分是好的，当且仅当：

- 所有区间被划分成**恰好两个非空子集**。
- 对于两个有公共点的区间，它们在同一个子集中。

tourist打算删掉若干个区间（可能一个也不删），使的存在一个好的划分。tourist 想知道最少要删掉多少个区间。

【输入格式】

从文件 *interval.in* 中读入数据。

本题有多组测试数据。

第一行一个整数 T ，表示数据组数。

每组数据的第一行一个整数 n 。

接下来 n 行，第 i 行包含两个整数 l_i, r_i 。

【输出格式】

输出到文件 *interval.out* 中。

对于每组数据输出一行一个整数表示答案。若无解，则输出 -1 。

【样例 1 输入】

```
5
2
3 7
8 9
3
1 4
2 6
5 7
2
1 4
4 6
10
7 94
58 74
15 42
27 34
58 73
20 96
1 15
42 95
58 84
51 94
9
10 79
```

```
11 84
33 82
4 66
19 32
20 23
16 89
12 96
54 61
```

【样例 1 输出】

```
0
1
-1
2
5
```

【样例 1 解释】

对于第二组数据，删除第 2 个区间，剩下两个区间各分为一个集合。

对于第三组数据，可以证明不存在合法解。

【样例 2】

见选手目录下的 *interval/interval2.in* 和 *interval/interval2.ans*。

该样例满足子任务 2 的限制。

【样例 3】

见选手目录下的 *interval/interval3.in* 和 *interval/interval3.ans*。

该样例满足子任务 3 的限制。

【样例 4】

见选手目录下的 *interval/interval4.in* 和 *interval/interval4.ans*。

该样例满足子任务 4 的限制。

【数据范围】

对于所有数据，保证 $1 \leq T \leq 6, 1 \leq n \leq 2 \times 10^5, 1 \leq l_i \leq r_i \leq 10^9$ 。

子任务编号	$n \leq$	$r_i \leq$	分值
1	18	10^6	10
2	1000	10^9	20
3	2×10^5	10^6	20
4	2×10^5	10^9	50

2.游客与染色 (color)

【题目描述】

tourist 有一个 $n \times m$ 的矩阵，他要将其填入 0 和 1 两种颜色。

我们称一个矩阵是好的，当且仅当满足以下条件：

- 每个点都有颜色。
- 对于每个 2×2 的子矩阵，颜色 0 的格子个数为偶数。

矩阵初始为全为无色。tourist 现在要进行 q 次如下操作：

- `x y c`：表示将矩阵第 x 行 y 列的格子涂上颜色 c ，**保证当前格子之前没有涂色。**

在每次操作结束后以及第一次操作前。你需要帮 tourist 计算出，如果对未被染色的格子染色，有多少染色方案是好的，答案对 $10^9 + 7$ 取模。

【输入格式】

输入文件名为 `color.in`。

第一行三个正整数 n, m, q ，表示矩阵的行数，列数和操作数。

接下来 q 行，每行 3 个数 `x y c` 表示每次操作。

【输出格式】

输出文件名为 `color.out`。

输出 $q + 1$ 行，表示初始时和每次操作后的答案。

【样例输入1】

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

【样例输出1】

```
8
4
2
1
0
```

【样例1解释】

在第 2 个操作后，满足条件的 2 个矩阵为：

$$\begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} \text{ 和 } \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}。$$

在第 4 个操作后，所有位置都已被涂色，但此时矩阵明显不是好的，所以输出 0。

【样例2】

见选手目录下的 *color/color2.in* 和 *color/color2.ans*。

该样例满足子任务 1 的限制。

【样例3】

见选手目录下的 *color/color3.in* 和 *color/color3.ans*。

该样例满足子任务 3 的限制。

【样例4】

见选手目录下的 *color/color4.in* 和 *color/color4.ans*。

该样例满足子任务 4 的限制。

【样例5】

见选手目录下的 *color/color5.in* 和 *color/color5.ans*。

该样例满足子任务 5 的限制。

【数据范围】

对于所有数据，保证

$1 \leq n, m \leq 3 \times 10^5, 0 \leq q \leq \min(nm, 5 \times 10^5), 1 \leq x \leq n, 1 \leq y \leq m, c \in \{0, 1\}$ 。

子任务编号	$n, m \leq$	$q \leq$	分值
1	4	10	10
2	3×10^5	0	20
3	9	30	15
4	1000	3000	15
5	3×10^5	5×10^5	40

3.游客与波特 (bot)

【题目描述】

bot 有一个 $1 \sim n$ 的排列 p 。

tourist 想知道排列 p ，他可以选择一个整数 $k(k \geq 0)$ ，并选择 k 个区间 $\{(l_1, r_1), (l_2, r_2), \dots, (l_k, r_k)\}$ 发送给 bot，其中 (l_i, r_i) 必须两两不同。

bot 会返回 k 个三元组 $\{(l_1, r_1, v_1), (l_2, r_2, v_2), \dots, (l_k, r_k, v_k)\}$ ，其中 $v_i = \max_{j=l_i}^{r_i} p_j$ ，表示每个区间的最大值。

tourist 想知道所有的 $2^{\frac{n(n+1)}{2}}$ 中选择方式中，有多少种能够根据 bot 的信息还原出排列 p 。由于答案很大，tourist 只需要你告诉他答案模 998244353 的结果。

由于 bot 是机器人，不会造数据，所以排列 p 一定满足一些特殊性质，具体请见【数据范围】。

【输入格式】

从文件 *bot.in* 中读取数据。

本题有多组测试数据。

第一行一个整数 T ，表示数据组数。

每组数据的第一行一个整数 n 。

接下来一行 n 个数，表示排列 p 。

【输出格式】

输出到文件 *bot.out* 中。

对于每组数据，输出一个数表示答案。

【样例 1 输入】

```
8
2
1 2
3
3 1 2
4
1 4 2 3
5
1 4 3 2 5
5
4 3 1 2 5
5
2 5 4 3 1
4
1 2 3 4
1
1
```

【样例 1 输出】

```
6
42
532
16702
19716
15508
630
2
```

【样例 1 解释】

该样例不满足题目数据范围和性质，选手仅供参考。

对于第一组数据，合法地选择有：

$\{(1, 1)\}, \{(2, 2)\}, \{(1, 1), (2, 2)\}, \{(1, 2), (2, 2)\}, \{(1, 1), (1, 2)\}, \{(1, 1), (1, 2), (2, 2)\}$ 。

对于第二组数据，一种可行方案是 $\{(2, 2), (2, 3)\}$ 。

对于第三组数据，一种可行方案是 $\{(2, 4), (3, 3), (3, 4)\}$ 。

对于第六组数据，一种可行方案是 $\{(1, 1), (1, 2), (1, 3), (1, 4)\}$ 。

注意，空集也可能成为合法方案。

【样例 2】

见选手目录下的 *bot/bot2.in* 和 *bot/bot2.ans*。

该样例满足子任务 2 的限制。

【样例 3】

见选手目录下的 *bot/bot3.in* 和 *bot/bot3.ans*。

该样例满足子任务 3 的限制。

【样例 4】

见选手目录下的 *bot/bot4.in* 和 *bot/bot4.ans*。

该样例满足子任务 4 的限制。

【样例 5】

见选手目录下的 *bot/bot5.in* 和 *bot/bot5.ans*。

该样例满足子任务 6 的限制。

【样例 6】

见选手目录下的 *bot/bot6.in* 和 *bot/bot6.ans*。

该样例满足子任务 8 的限制。

【数据范围】

对于所有数据，保证 $1 \leq T \leq 3, 1 \leq n \leq 5 \times 10^5$ ， p 为排列且满足 A, B, C, D 四种特殊性质之一。

子任务编号	$n \leq$	特殊性质	分值
1	5	D	15
2	2000	A	10
3	2000	B	20
4	2000	C	15
5	300	D	10
6	2000	D	10
7	10^5	D	15
8	5×10^5	D	5

特殊性质 A：保证 $p_i = i$ 。

特殊性质 B：设 x 满足 $p_x = 1$ ，则对于 $1 \leq i < x$ ，满足 $p_i > p_{i+1}$ ，对于 $x \leq i < n$ ，满足 $p_i < p_{i+1}$ 。

特殊性质 C：设 x 满足 $p_x = n$ ，则对于 $1 \leq i < x$ ，满足 $p_i < p_{i+1}$ ，对于 $x \leq i < n$ ，满足 $p_i > p_{i+1}$ 。

特殊性质 D：保证 p 从所有长度为 n 的排列中**随机均匀选择**。

4.游客与串串 (string)

【题目描述】

tourist 心血来潮, 准备制作一个文本编辑器。但他要忙着冲击 CF4000 分, 这个任务就交给你了。

在文本编辑器中, 你需要帮 tourist 维护一个长度为 n 的字符串 c 。

tourist 还有一个包含 m 个**单词**的字典, 每个单词都是一个字符串, 称第 i 个单词为 s_i 。

文本编辑器有以下功能:

- **查找功能**: 有两个参数 l, r , 表示询问对于字典中的每个**单词** s_i , $c[l:r]$ 中 s_i 的出现次数之和。
- **替换功能**: 有三个参数 l, r, t , 其中 t 是一个字符串, 表示将 $c[l:r]$ 替换为 t 不断重复的结果。示例: 把 botlovebot 替换为 hqj 不断重复的结果为 hqjhqjhqjh。

tourist 给出了 q 个操作, 每个操作是两个实现功能之一, 你需要正确回答每个**查找功能**的结果。

其中, 对于字符串 s , 符号 $s[l:r](1 \leq l \leq r \leq |s|)$ 表示 s 中从 l 到 r 的**子串**。

【输入格式】

输入文件名为 `string.in`。

第一行三个整数 n, m, q , 依次表示文件 c 的长度, 字典中的单词数, 询问的个数。

第二行一个长度为 n 的字符串 c , 表示初始文件。

接下来 m 行, 第 i 行一个字符串, 表示第 i 个单词 s_i 。

接下来 q 行, 每行表示一个操作:

每行的第一个数 op 表示此次操作的类型;

若 $op = 1$, 则接下来两个整数 l, r , 表示这是一次**查找**;

若 $op = 2$, 则接下来两个整数 l, r 和一个字符串 t , 表示这是一次**替换**。

【输出格式】

输出文件名为 `string.out`。

对于每个**查找**操作, 一行一个整数, 表示本次操作的结果。

【样例输入1】

```
6 2 5
bbabba
bb
bab
1 1 6
2 3 5 a
1 2 3
2 1 6 b
1 1 5
```

【样例输出1】

```
3
0
4
```

【样例1解释】

第一次操作时 s_1 的出现次数为 2, s_2 的出现次数为 1。

第二次操作后 c 变为 `bbaaaa`。

第三次操作时 s_1, s_2 均未出现。

第四次操作后 c 变为 `bbbbbb`。

第五次操作时 s_1 的出现次数为 4, s_2 未出现。

【样例2】

见选手目录下的 `string/string1.in` 和 `string/string1.ans`。

该样例满足子任务 1 的限制。

【样例 3】

见选手目录下的 `string/string2.in` 和 `string/string2.ans`。

该样例满足子任务 2 的限制。

【样例 4】

见选手目录下的 `string/string3.in` 和 `string/string3.ans`。

该样例满足子任务 3 的限制。

【样例 5】

见选手目录下的 `string/string4.in` 和 `string/string4.ans`。

该样例满足子任务 4 的限制。

【样例 6】

见选手目录下的 `string/string5.in` 和 `string/string5.ans`。

该样例满足子任务 5 的限制。

【样例 7】

见选手目录下的 `string/string6.in` 和 `string/string6.ans`。

该样例满足子任务 6 的限制。

【数据范围】

对于所有数据, 保证 $1 \leq l \leq r \leq n \leq 10^6$, $1 \leq m, q \leq 10^5$, $op \in \{1, 2\}$, $|s_i| \leq 50$, $\sum |s_i| \leq 2 \times 10^5$, $\sum |t| \leq 10^6$, 所有字符串保证不为空串, 且出现的字符均为小写字母。

注意: 单词的长度非常小

子任务编号	$n \leq$	$q \leq$	$\sum s_i \leq$	$\sum t \leq$	特殊性质	分值
1	100	100	100	10^6	无	10
2	3000	3000	3000	10^6	无	10
3	10^6	10^5	50	10^6	$m = 1$	20
4	10^6	10^5	2×10^5	10^6	没有替换操作	20
5	5×10^4	5×10^4	5×10^4	5×10^4	无	20
6	10^6	10^5	2×10^5	10^6	无	20