

2024年NOIP模拟赛

时间：2024 年 11 月 01 日

by GDSY

题目名称	草莓	三色	博弈	后缀数组
题目类型	传统型	传统型	传统型	传统型
目录	guiltiness	color	game	sa
可执行文件名	guiltiness	color	game	sa
输入文件名	guiltiness.in	color.in	game.in	sa.in
输出文件名	guiltiness.out	color.out	game.out	sa.out
每个测试点时限	2s	1s	3s	2s
内存限制	1024MB	1024MB	1024MB	1024MB
是否捆绑测试	否	是	是	是

提交源程序文件名

对于 C++ 语言	desire.cpp	dealing.cpp	lunatic.cpp	season.cpp

编译选项

```
-lm -std=c++14 -o2 -w1,--stack=998244353
```

草莓(guiltiness)

题目描述

Madeline 正在吃草莓，但草莓太大了，她决定把草莓掰开带走。

草莓有 n 行 m 列，一开始是一整块。

Madeline 有两种操作：

1. 横着掰：将草莓的第 i 行和第 $i + 1$ 行掰开，花费为 $x_i \times (\text{已经竖着掰的次数} + 1)$ 。
2. 竖着掰：将草莓的第 i 列和第 $i + 1$ 列掰开，花费为 $y_i \times (\text{已经横着掰的次数} + 1)$ 。

Madeline 想知道把草莓全部掰成 1×1 的小块最少需要多长时间。

输入格式

从 `guiltiness.in` 中读入数据。

第一行包括两个整数 n, m 。

第二行包含 $n - 1$ 个整数 x_i 。

第三行包含 $m - 1$ 个整数 y_i 。

输出格式

输出到 `guiltiness.out` 中。

一个整数表示最短时间。

输入输出样例

输入样例 1

```
1 | 6 4
2 | 2 1 3 1 4
3 | 4 1 2
```

输出样例 1

```
1 | 42
```

数据范围

对于所有数据： $1 \leq n, m \leq 2 \times 10^5, 1 \leq x_i, y_i \leq 2 \times 10^5$ 。

对于 20% 的数据， $1 \leq n, m \leq 5$ 。

对于 60% 的数据， $1 \leq n, m \leq 5 \times 10^3$ 。

对于另外 10% 的数据，所有 x_i 相同，所有 y_i 相同。

三色(color)

题目描述

Madeline 的头发有三种颜色——红色、蓝色和紫色。

现在有 n 个格子排成一排，从左至右编号为 1 至 n ，你要把这些格子分别染成三色中的一种颜色。

有 m 个限制，第 i 个限制形如：第 l_i 到第 r_i 个格子内**恰好**有 x_i 种颜色。

求方案数对 $10^9 + 7$ 取模的结果。

输入格式

从 `color.in` 中读入数据。

第一行一个整数 T 表示数据组数。

对于每组数据，第一行两个整数 n, m 。

接下来 m 行，第 i 行三个整数 l_i, r_i, x_i 。

输出格式

输出到 `color.out` 中。

对于每组数据，输出一个整数表示答案。

输入输出样例

输入样例 1

```
1 4
2 1 0
3 2 0
4 3 0
5 5 2
6 1 3 3
7 4 5 1
```

输出样例 1

```
1 3
2 9
3 27
4 18
```

数据范围

对于所有数据： $\sum n \leq 5000, \sum m \leq 10^6, 1 \leq x_i \leq 3, 1 \leq l_i \leq r_i \leq n$ 。

子任务 1 (10 分)： $\sum n \leq 100$ 。

子任务 2 (15 分)： $\sum n \leq 500$ 。

子任务 3 (15 分)：无特殊限制。

博弈 (game)

题目描述

Madeline 和 Badeline 在玩游戏。

这里有一个长度为 n 的数组 a ，两人轮流行动，Madeline 先手。

在每一轮，当前的玩家要选择两个数 $a_i, a_j (i \neq j)$ ，然后改变它们的值。假设改变之后的数值为 a'_i, a'_j ，你需要保证 $a_i + a_j = a'_i + a'_j$ 和 $|a'_i - a'_j| < |a_i - a_j|$ 。无法行动者输。

在游戏开始之前，Madeline 要将 a 删至只剩三个元素。请你求出 Madeline 有多少操作方式使得她有必胜策略。

两种操作方式不同当且仅当存在 i ，使得 a_i 在一种操作方式中被保留，在另外一种操作方式中未被保留。

输入格式

从 `game.in` 中读入数据。

第一行给定一个整数 T ，表示数据组数。

接下来对于每组数据，第一行给定一个整数 n ，第二行给定 n 个数 $a_1 \dots a_n$ 。

输出格式

输出到 `game.out` 中。

对于每组数据，输出一个整数表示答案。

输入输出样例

样例输入 1

```
1 3
2 4
3 2 0 2 3
4 3
5 2 2 3
6 3
7 0 2 3
```

样例输出 1

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

解释

对于第一组数据，只有删掉 a_2 会使 Madeline 输掉游戏。

数据范围

$3 \leq n \leq 5 \times 10^5, \sum n \leq 3 \times 10^6, 0 \leq a_i \leq 10^{18}$ 。

子任务 1 (10 分) : $T = 1, n, a_i \leq 5$ 。

子任务 2 (15 分) : a_i 互不相同。

子任务 3 (15 分) : $\sum n \leq 400, a_i \leq 10^5$ 。

子任务 4 (35 分) : $\sum n \leq 3 \times 10^5$ 。

子任务 5 (25 分) : 无特殊限制。

后缀数组 (sa)

题目描述

定义**字符串**为一个由正整数组成的序列，且序列的最大值等于序列的数的种类数。例如 2, 1, 1 和 1, 2, 4, 3, 3 是字符串，而 1, 3, 3 不是。

定义一个**字符串的后缀数组**为所有后缀按照字典序排序后的起始点顺序，也就是说长度为 n 的串 s 的后缀数组 a 是一个满足 $s[a_{i-1} : n] < s[a_i : n]$ 的长度为 n 的排列。

Madeline 有一个字符串的后缀数组，但她不知道原字符串是什么，她想请你帮她求出原字符串有多少种可能，答案对 998244353 取模。

由于后缀数组的长度太长了，我们使用如下方式输入：初始化一个长度为 n 的数组 a ，一开始 $a_i = i$ ，接下来进行 m 次操作。操作分为两种：

- **0 u v**：将 $a[u : v]$ 移到开头，操作之后 a 形如 $a_u, a_{u+1} \dots a_v, a_1, a_2 \dots a_{u-1}, a_{v+1} \dots a_n$ 。
- **1 u v**：将 $a[u : v]$ 翻转，操作之后 a 形如 $a_1, a_2 \dots a_{u-1}, a_v, a_{v-1} \dots a_u, a_{v+1} \dots a_n$ 。

输入格式

从 `sa.in` 中读入数据。

第一行给定两个整数 n, m 。

接下来 m 行输入 3 个整数 op, u, v ， $op \in \{0, 1\}, 1 \leq u \leq v \leq n$ 。

输出格式

输出到 `sa.out` 中。

输出一个整数表示答案。

样例输入输出

样例输入 1

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

样例输出 1

```
1 4
```

数据范围

对于所有数据： $n \leq 10^9, m \leq 10^5$ 。

子任务 1 (10分)： $n \leq 8$ 。

子任务 2 (20分)： $n, m \leq 1000$ 。

子任务 3 (30分)： $n \leq 10^5$ 。

子任务 4 (40分)：无特殊限制。

