

CCF 全国青少年信息学奥林匹克联赛

CCF NOIP 2024

模拟试题

时间：2024年10月02日 8:00 ~ 12:30

题目	移动	子串结构	简单博弈	岭上开花
题目类型	传统型	传统型	传统型	传统型
目录	move	string	game	que
可执行文件名	move	string	game	que
输入文件名	move.in	string.in	game.in	que.in
输出文件名	move.out	string.out	game.out	que.out
每个测试点时限	1.0秒	1.0秒	1.0秒	2.0秒
内存限制	512 MiB	512 MiB	512 MiB	1024 MiB
测试点数目	10	20	20	20
测试点是否等分	是	是	是	是

提交源程序文件名

对于C++ 语言	move.cpp	string.cpp	game.cpp	que.cpp
----------	----------	------------	----------	---------

编译选项

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

注意事项（请仔细阅读）

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C/C++ 中函数main()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。
3. 提交的程序代码文件的放置位置请参考各省的具体要求。
4. 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。
5. 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）。
6. 选手提交的程序源文件必须不大于100KB。
7. 程序可使用的栈空间内存限制与题目的内存限制一致。
8. 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。
9. 只提供Linux格式附加样例文件。
10. 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此为准。

移动 (move)

【题目描述】

数轴上有 n 个点，小 A 初始在原点。

在每一步，小 A 可以选择一个点，如果小 A 的位置不与该点重合，则向它的方向移动一个单位长度，反之加 1 分。每个点在被选择后消失。

请帮助小 A 设计一个策略，使得分数最大化。

【输入格式】

从文件 `move.in` 中读入数据。

每一组数据内包含多个测试点。

数据的第一行为一个正整数 T ，代表测试点个数。

每个测试点的第一行包含一个正整数 n ，代表数轴上点的个数。

每个测试点的第二行包含 n 个正整数 a_i ，代表每个点的坐标。

【输出格式】

输出到文件 `move.out` 中。

对于每一个测试点，输出一行包含一个正整数，代表小 A 的最大得分。

【样例1输入】

```
1 | 1
2 | 10
3 | -6 -2 -2 -1 -1 -1 1 2 6 8
```

【样例1输出】

```
1 | 5
```

【样例1解释】

小 A 按照 $\{-6, -1, -1, -1, -2, -2, 8, 6, 2, 1\}$ 的顺序选点，每次选点后的坐标分别为 $\{-1, -1, -1, -1, -2, -2, -1, 0, 1, 1\}$ ，得分为 $\{0, 1, 2, 3, 3, 4, 4, 4, 4, 5\}$ ，可以证明这是最优的策略。

【样例2】

见选手目录下的 `move/move2.in` 与 `move/move2.ans`。

【数据范围】

对于所有数据，有 $1 \leq n \leq 10^5, 1 \leq T \leq 100, \sum n \leq 3 \times 10^5, |a_i| \leq 10^5$ 。

保证所有 a_i 按升序排列。

测试点	$\sum n \leq$
1 ~ 2	20
3 ~ 5	10^3
6 ~ 10	3×10^5

子串结构 (string)

【题目描述】

给定两个由小写字母构成的字符串 A, B ，需要构造一个由小写字母构成的字符串 C ，使得 A, B 在 C 中的出现次数相等且不为 0。

如果存在构造方案，请输出 C 的长度的最小值，反之输出 -1 。

【输入格式】

从文件 `string.in` 中读入数据。

每一组数据内包含多个测试点。

第一行一个正整数 T ，代表测试点个数。

每个测试点一行，包含两个字符串 A, B ，表示题述字符串。

【输出格式】

输出到文件 `string.out` 中。

对每一个测试点输出一行整数，代表答案。

【样例1输入】

```
1 3
2 ab ba
3 a aaa
4 ab aba
```

【样例1输出】

```
1 3
2 -1
3 3
```

【样例1解释】

对于第一个测试点和第三个测试点，可以构造 $C = aba$ 满足 A, B 各出现一次。

可以证明第二个测试点无解。

【样例2】

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

【数据范围】

对于所有数据，有 $1 \leq T \leq 10^5, \sum |A|, \sum |B| \leq 10^6$ 。

测试点	$\sum A , \sum B \leq$	特殊性质
1 ~ 5	10^3	字符串仅含字母 a, b

测试点	$\sum A , \sum B \leq$	特殊性质
6 ~ 10	10^6	字符串仅含字母 a, b
11 ~ 20	10^6	无

简单博弈 (game)

【题目描述】

n 个人排成一排，而小 B 要从左往右和他们依次进行石头剪刀布的游戏。

在游戏开始前，小 B 会预测这 n 个人的决策，并将其视作一个长为 n 的序列 a 。其中 $a_i = 0, 1, 2$ 分别代表第 i 个人会出石头、布或者剪刀。为了增加游戏的挑战性，小 B 不能在和两场连续的游戏出相同的手势，即如果小 B 与第 i 个人游戏时出了剪刀，则他与第 $i + 1$ 个人游戏时就不能出剪刀。

在这 n 场游戏之后，小 B 的得分可以计算如下：

- 如果小 B 赢下了某一场游戏，则加 1 分
- 如果小 B 在某一场比赛中平局，则不加分
- 如果小 B 输掉了某一场游戏，则扣 $10^{999999999999999999}$ 分

由于某些人的决策很好预测，而另一些人较为善变，所以小 B 在思考后给出了一个序列 b ，其中 $b_i \in [-1, 0, 1, 2]$ 。当 $b_i = -1$ 时代表第 i 个人的决策为 $\{0, 1, 2\}$ 的概率是相等的，反之则代表第 i 个人的决策是固定为 $\{0, 1, 2\}$ 其中之一。

记 b 中 -1 的个数为 k ，则可能的预测序列 a 的个数为 3^k 。请对这 3^k 个序列求出每个序列小 B 在最优决策下的得分之和。结果对 998244353 取模。

【输入格式】

从文件 `game.in` 中读入数据。

第一行包含一个正整数 n ，代表人数。

第二行包含 n 个 $[-1, 2]$ 间的整数 b_i ，代表题中序列 b 的每个元素的值。

【输出格式】

输出到文件 `game.out` 中。

输出一行包含一个正整数，代表最后的得分之和。

【样例1输入】

```
1 | 4
2 | -1 0 1 2
```

【样例1输出】

```
1 | 11
```

【样例1解释】

若 $a = \{0, 0, 1, 2\}$ ，则小 B 的最优决策为 $\{1, 0, 2, 0\}$ ，得分为 3。

若 $a = \{1, 0, 1, 2\}$ ，则小 B 的最优决策为 $\{2, 1, 2, 0\}$ ，得分为 4。

若 $a = \{2, 0, 1, 2\}$ ，则小 B 的最优决策为 $\{0, 1, 2, 0\}$ ，得分为 4。

最终结果为 $3 + 4 + 4 = 11$

【样例2】

见选手目录下的 `game/game2.in` 与 `game/game2.ans` 。

【数据范围】

对于所有数据, 有 $1 \leq n \leq 2 \times 10^3, -1 \leq b_i \leq 2$ 。

测试点	$n \leq$	特殊性质
1 ~ 4	300	满足 $b_i = -1$ 的 i 的个数小于 10
5 ~ 8	2×10^3	所有 $b_i = -1$
9 ~ 16	300	无
17 ~ 20	2×10^3	无

岭上开花 (que)

【题目描述】

小 D 在玩一种回合制游戏，当前他面对的敌人血量为 h ，敌人不会进行攻击。战斗方式如下：

牌堆里有无限张牌，花色编号为 $\{1, 2, 3\}$ ，战斗开始时手牌为空，小 D 操控的角色每回合会进行如下行动：

- 抽 2 张牌，若当前手牌数量为 3，则只抽一张牌
- 如果当前持有 4 张花色相同的手牌，则全部弃掉并造成 a 点伤害，反之弃掉花色数量最小的一张手牌（若相同则弃掉花色编号最小的一张手牌）并造成 b 点伤害。注意弃牌不返回牌堆。

服务器中有一个长为 n 的花色序列，每场战斗选择其中一个连续区间 $[l, r]$ **无限循环** 作为当前的抽牌堆，即小 D 抽到的第 k 张牌是原序列的第 $l + [k \bmod (r - l + 1)]$ 张。

服务器的花色序列是动态的，有 m 次操作，分为两种：

- 将序列中 x 位置的牌的花色修改为 y
- 询问选择区间 $[l, r]$ 作为抽牌堆时，在敌人血量为 h ，两种弃牌方式对应的伤害为 a, b 时，需要多少回合可以击败敌人

【输入格式】

从文件 `que.in` 中读入数据。

输入的第一行包含两个正整数 n, m ，代表牌堆长度与询问数量。

接下来一行包含 n 个正整数 a_i ，代表花色序列。

接下来 m 行每行一个操作，格式如下：

- `1 x y`：将序列中 x 位置的牌的花色修改为 y
- `2 l r h a b`：询问选择区间 $[l, r]$ 作为抽牌堆时，在敌人血量为 h ，两种弃牌方式对应的伤害为 a, b 时，需要多少回合可以击败敌人

【输出格式】

输出到文件 `que.out` 中。

对于每个 2 操作，输出击败敌人的回合数。

【样例1输入】

```
1 | 5 1
2 | 1 3 3 1 2
3 | 2 2 4 10 2 1
```

【样例1输出】

```
1 | 8
```


【样例1解释】

抽牌堆为 $\{3, 3, 1, 3, 3, 1, 3, 3, 1, 3, 3, 1, \dots\}$ 。

第一回合手牌为 $(3, 3)$ ，弃掉 3，造成 1 点伤害。

第二回合手牌为 $(3, 1, 3)$ ，弃掉 1，造成 1 点伤害。

第三回合手牌为 $(3, 3, 3, 1)$ ，弃掉 1，造成 1 点伤害。

第四回合手牌为 $(3, 3, 3, 3)$ ，全部弃掉，造成 2 点伤害。

第五回合手牌为 $(3, 1)$ ，弃掉 1，造成 1 点伤害。

第六回合手牌为 $(3, 3, 3)$ ，弃掉 3，造成 1 点伤害。

第七回合手牌为 $(3, 3, 1, 3)$ ，弃掉 1，造成 1 点伤害。

第八回合手牌为 $(3, 3, 3, 3)$ ，全部弃掉，造成 2 点伤害，敌人死亡。

【样例2】

见选手目录下的 `que/que2.in` 与 `que/que2.ans`。

【数据范围】

对于所有数据，有 $1 \leq n, m \leq 10^5, 1 \leq h \leq 10^{13}, 1 \leq a, b \leq 10^9$
 $, 1 \leq x, l, r \leq n, y, a_i \in \{1, 2, 3\}$

测试点	特殊性质
1 ~ 3	$1 \leq n, m \leq 500, y \in \{1, 2\}$
4 ~ 7	$y \in \{1, 2\}$
8 ~ 14	保证不存在修改操作
15 ~ 20	无