

省选 2024 模拟赛 Day 1 (2.16)

By YeahPotato

2024.2.16

题目名称	染色数组	博弈	消消乐
题目类型	传统型	传统型	传统型
源程序文件名	color.cpp	game.cpp	bubbles.cpp
输入文件名	color.in	game.in	bubbles.in
输出文件名	color.out	game.out	bubbles.out
每个测试点时限	1s	1s	1s
内存限制	7~512MB	256MB	1GB
子任务数目	4	7	9
是否有 spj	否	否	否

开 O2、无限栈、C++ 11 以上。

评测机上的单个测试点时限为上表规定时限与 std 在本题测试点中运行时间最大值的 1.5 倍的较大值。

染色数组

题目描述

YeahPotato 稍微改了一下联合省选 2023 D2T3 的题面。

给定一个长度为 n 的正整数数组 a_1, \dots, a_n , 从左到右排成一排。现在要将每个数字染成红色或者绿色, 我们定义一个染色方案为优秀的染色方案, 当且仅当它满足:

- 1. 每个数 a_i 要么被染成红色, 要么被染成绿色。
- 2. 红色的数从左到右依次递增, 绿色的数从左到右依次递增。这里, 递增指**非严格递增**。

例如: 1 6 3 4 7 9 中, 将 1 3 4 7 染成红色, 6 9 染成绿色是优秀的染色方案 (1**6**3**4**7**9**) ; 1 3 4 9 染成红色, 6 7 染成绿色也是优秀的染色方案 (1**6**3**4**7**9**) 。但是将 1 4 7 9 染成红色, 6 3 染成绿色则**不是**优秀的染色方案, 因为 6 3 不是递增的。1 9 5 5 中, 将 1 9 染成红色, 5 5 染成绿色, 也是优秀的染色方案 (1**9****5****5**) 。

求给定数组有多少种优秀的染色方案, 答案对 998244353 取模。

输入格式

本题有多组数据。第一行一个正整数 T , 表示数据组数。

每组数据第一行一个正整数 n , 第二行 n 个正整数 a_1, \dots, a_n 。

输出格式

T 行依次表示每组数据的答案。

样例输入 1

```
4
2
2 1
2
1 2
4
2 4 1 3
12
1 1 4 5 1 4 1 9 1 9 8 10
```

样例输出 1

```
2
4
2
0
```

样例解释 1

四组数据的优秀染色方案分别为：

1. 21、21
2. 12、12、12、12
3. 2413、2413
4. 无

样例 2

见于下发文件中。

数据范围

Subtask 1(20%): $T \leq 100$, $n \leq 18$ 。

Subtask 2(25%): $\sum n \leq 5000$ 。

Subtask 3(25%): $\sum n \leq 3 \times 10^5$ 。

Subtask 4(30%): $\sum n \leq 10^6$ 。

对于 100% 的数据, $1 \leq T \leq 1000$, $1 \leq n \leq 10^6$, $1 \leq a_i \leq 10^9$ 。

特别地, Subtask 1,2,3 的空间限制为 512MB, Subtask 4 的空间限制为 7MB。

提示

本题输入量较大, 建议使用较快的读入方式。

请务必考虑程序本身占用的内存空间。

博弈

题目描述

YeahPotato 稍微改了一下【数据删除】的题面。

假设足够聪明的 YeahPotato 和 NoPotato 在玩一个博弈游戏。

有一棵 n 个点的树，点编号 $1, \dots, n$ ，边无向。点 i 上面放有 c_i 颗棋子。两人轮流行动，YeahPotato 先手。当前行动者需要选一颗棋子移到相邻的一个点，并且其会获得新落点编号的分数。棋子可以移到之前到过的点，也可以多颗棋子重合。在第 114^{514} 轮后，如果 YeahPotato 的得分和多于 NoPotato，则 YeahPotato 获胜。

游戏开始前，YeahPotato 可以选择任意数量的边切断，但要保证使至少一个棋子能移动。他想知道，有多少种选择方法，能使他获胜。

然而 YeahPotato 并不足够聪明，所以他希望你帮他求出答案对 998244353 取模后的值。

输入格式

第一行一个整数 n 。

第二行 n 个整数 c_1, \dots, c_n 。

接下来 $n - 1$ 行，每行两个整数 u, v ，表示一条边。保证构成一棵树。

输出格式

一行一个答案。

样例输入 1

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

样例输出 1

```
5
```

样例输入 2

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

样例输出 2

77

样例 3

见于下发文件中。

数据范围

子任务编号	$n \leq$	特殊性质	分值
1	8	A	10
2	18		10
3	1.5×10^5	B	10
4	1.5×10^5	A	10
5	10^3		20
6	5×10^4		15
7	1.5×10^5		25

特殊性质 A: $\sum_{i=1}^n c_i = 1$ 。

特殊性质 B: 树为一条按顺序编号的链, 即 $E = \{(i, i + 1) \mid 1 \leq i < n\}$ 。

对于 100% 的数据, $2 \leq n \leq 1.5 \times 10^5, 0 \leq c_i \leq 10, \sum_{i=1}^n c_i \geq 1$ 。

消消乐

题目背景

Emacs > bubbles > hard

题目描述

YeahPotato 稍微改了一下 CSP-S 2023 T2 的题面。

现在，他有一个长度为 n 的 01 串。我们称一个 01 串是可消除的，当且仅当可以对这个串进行若干次操作，使之成为一个空串。其中每次操作可以从串中删除**两个或三个连续**的相同位，操作后剩余两侧串会拼接在一起。

他想让你算出，这个 01 串的所有非空子串中，有多少个是可消除的。子串指连续的子序列。

输入格式

一行一个 01 串。

输出格式

一行一个答案。

样例输入 1

10110

样例输出 1

2

样例解释 1

可消除的区间有 $[3, 4] : \underline{11} \rightarrow \emptyset$ 以及 $[2, 5] : 0\underline{110} \rightarrow \underline{00} \rightarrow \emptyset$ 。 \emptyset 表示已经消光的空串。

样例输入 2

0101000

样例输出 2

3

样例解释 2

可消除的区间有 $[5, 6], [6, 7], [5, 7]$ 。

样例输入 3

101100010110

样例输出 3

22

样例解释 3

原串本身是可消除的：101100010110 → 1011010110 → 10010110 → 110110 → 1100 → 00 → ∅。

样例输入 4

000100010100010100010100000110010001100110001010

样例输出 4

834

数据范围

子任务编号	$n \leq$	分值
1	10	5
2	100	10
3	100	5
4	500	20
5	2000	10
6	8000	5
7	200000	20
8	2000000	10
9	10000000	15

特殊性质：对于子任务 2，最长的不可消除子串长至多为 20。

对于 100% 的数据， $1 \leq n \leq 10^7$ 。

提示

由于输入量较大，建议在子任务 8,9 中使用较快的读入方式。假设你希望将输入存储到 char 数组 S 的 1 到 n 位置中，可以使用如下代码。你可以根据自己的需要调整 fread 前后的操作。

```
char S[10000007]; int n = 0;
fread (S + 1, 1, 1e7, stdin);
while (isdigit(S[n+1])) n ++;
S[n+1] = 0;
```