

NOIP 模拟赛

第九试

时间：2024 年 11 月 27 日 08:00 ~ 12:30

题目名称	四种颜色	四倍快乐	四进制数	四格方块
题目类型	传统型	传统型	传统型	传统型
目录	four	quadruple	quaternary	tetris
可执行文件名	four	quadruple	quaternary	tetris
输入文件名	four.in	quadruple.in	quaternary.in	tetris.in
输出文件名	four.out	quadruple.out	quaternary.out	tetris.out
每个测试点时限	1.0 秒	1.0 秒	4.0 秒	1.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
测试点数目	20	20	20	25
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	four.cpp	quadruple.cpp	quaternary.cpp	tetris.cpp
-----------	----------	---------------	----------------	------------

编译选项

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

四种颜色 (four)

【题目描述】

有一张分成 n 个格子的纸条。每个格子都被涂成了一种颜色。
你需要把这个纸条剪成若干段，使得每一段都恰好有四种不同的颜色。
求有多少种不同的方案。答案对 998244353 取模。

【输入格式】

从文件 *four.in* 中读入数据。
第一行一个正整数 n 。
第二行 n 个正整数，其中第 i 个数 c_i 表示从左到右第 i 个格子的颜色。

【输出格式】

输出到文件 *four.out* 中。
一行一个整数表示答案。

【样例 1 输入】

```
1 9
2 1 2 3 4 1 2 3 4 1
```

【样例 1 输出】

```
1 3
```

【样例 1 解释】

有三种方案。一种是不剪：(1, 2, 3, 4, 1, 2, 3, 4, 1)。一种是从第四格和第五格之间剪开：(1, 2, 3, 4), (1, 2, 3, 4, 1)。一种是从第五格和第六格之间剪开：(1, 2, 3, 4, 1), (2, 3, 4, 1)。

【样例 2】

见选手目录下的 *four/four2.in* 与 *four/four2.ans*。

【子任务】

保证 $1 \leq n \leq 300000$, $1 \leq c_i \leq n$ 。

对于 20% 的数据, 保证 $n \leq 20$ 。

对于另外 25% 的数据, 保证 $n \leq 2000$ 。

对于另外 25% 的数据, 保证 $1 \leq c_i \leq 4$ 。

对于剩下 30% 的数据, 没有特殊限制。

四倍快乐 (quadruple)

【题目描述】

定义“四倍快乐”串为形如 $AABAA$ 的字符串，其中 A 为任意非空字符串， B 为任意字符串。同一个字符串可能是多种形式的“四倍快乐”串，因为可能可以选择多种不同的 A 。给定字符串 S ，对 S 的每个前缀询问它同时是多少种形式的“四倍快乐”串。

【输入格式】

从文件 `quadruple.in` 中读入数据。
一行一个字符串 S 。 S 仅由小写字母构成。

【输出格式】

输出到文件 `quadruple.out` 中。

设 n 为 S 的长度。你本来应该输出 n 行每行一个整数，其中第 i 行表示长度为 i 的前缀同时是多少种形式的“四倍快乐”串。但是为了减少输出长度，设 $k = \lceil \frac{n}{10^5} \rceil$ ，其中 $\lceil x \rceil$ 表示大于等于 x 的最小整数。你需要把这 n 个数按顺序每 k 个分成一组，输出每组的和。即输出的第一行为原本的前 k 个数的和，第二行为原本的第 $k+1 \sim 2k$ 个数的和，以此类推。

【样例 1 输入】

```
1 lalawalala
```

【样例 1 输出】

```
1 0
2 0
3 0
4 0
5 0
6 0
7 0
8 0
9 0
10 1
```

【样例 1 解释】

只有全串 `lalawalala` 是一个“四倍快乐”串，唯一可以选择的 A 是 `la`。

【样例 2】

见选手目录下的 `quadruple/quadruple2.in` 与 `quadruple/quadruple2.ans`。

【样例 2 解释】

该样例中 $n = 999999$ 。

【样例 3】

见选手目录下的 `quadruple/quadruple3.in` 与 `quadruple/quadruple3.ans`。

【样例 3 解释】

该样例中 $n = 10^6$ 。

【子任务】

保证 $1 \leq n \leq 5 \times 10^6$ 。

对于 15% 的数据，保证 $n \leq 200$ 。

对于另外 25% 的数据，保证 $n \leq 2000$ 。

对于另外 40% 的数据，保证 $n \leq 2 \times 10^5$ 。

对于剩下 20% 的数据，没有特殊限制。

四进制数 (quaternary)

【题目描述】

你可以多次对一个四进制数 $n = \overline{d_c d_{c-1} \cdots d_1 d_0}$ 进行如下之一的变换：

- 选择一个 i 满足 $1 \leq i \leq c$ 且 $d_i < d_{i-1}$ 。交换 d_i 和 d_{i-1} 。
- 选择一个 i 满足 $2 \leq i \leq c$ 且 $d_i < d_{i-2}$ 。交换 d_i 和 d_{i-2} 。

你需要对每个 $1 \leq k \leq r$ 的 k 求出 n 通过若干次变换（可以是 0 次）后可以得到的第 k 小的数。

【输入格式】

从文件 *quaternary.in* 中读入数据。

第一行一个正整数 n 。

第二行一个正整数 r 。

【输出格式】

输出到文件 *quaternary.out* 中。

r 行，每行一个整数，其中第 i 行表示 $k = i$ 时的答案。若不存在输出 -1 。

【样例 1 输入】

```
1 18
2 5
```

【样例 1 输出】

```
1 18
2 24
3 33
4 36
5 -1
```

【样例 1 解释】

18 的四进制表示是 102。

不做变换可以得到 $(102)_4 = 18$ 。

交换最后两位可以得到 $(120)_4 = 24$ 。

交换第一位和第三位可以得到 $(201)_4 = 33$ 。

交换后两位再交换前两位，或者先交换第一位和第三位再交换后两位，可以得到 $(210)_4 = 36$ 。

一共只有这四个数能通过变换得到，所以不存在第五小的数。

【样例 2】

见选手目录下的 *quaternary/quaternary2.in* 与 *quaternary/quaternary2.ans*。

【子任务】

保证 $1 \leq n < 2^{60}$ 。保证 $1 \leq r \leq 10^6$ 。

对于 20% 的数据，保证 $n < 2^{20}$ 。

对于另外 20% 的数据，保证 $r \leq 800$ 。

对于另外 20% 的数据，保证 $r \leq 3000$ 。

对于另外 20% 的数据，保证 $r \leq 10^5$ 。

对于剩下 20% 的数据，没有特殊限制。

四格方块 (tetris)

【题目描述】

你需要用 n 个四格方块填满一个四行 n 列的矩形。
四格方块有下面这七种：

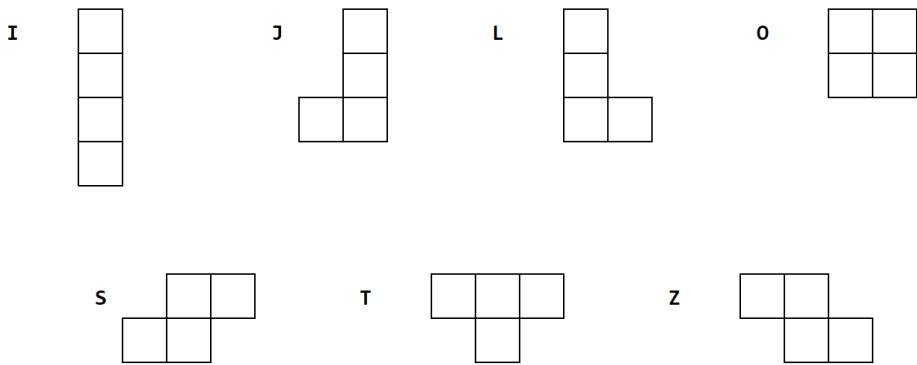


图 1: 七种四格方块

方块可以旋转放置但不能翻转。在这 $4 \times n$ 个格子中，有些格子有一个形状限制，也就是说，这个格子必须被某一种形状的四格方块覆盖。
求一共有多少种不同的放置方案。答案对 998244353 取模。

【输入格式】

从文件 `tetris.in` 中读入数据。
第一行一个正整数 n 。
接下来 4 行，一个 n 列的字符矩阵表示每个格子上的限制。其中 `IJLOSTZ` 中的字母表示对应的形状限制。`.` 表示没有限制。

【输出格式】

输出到文件 `tetris.out` 中。
一行一个整数，表示方案数模 998244353 的值。

【样例 1 输入】

```
1 10
2 .O...J.L..
3 ...S.....T
4 .L...O.Z..
```


5

...I.....T

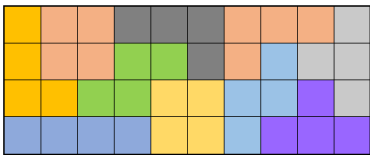
【样例 1 输出】

1

1

【样例 1 解释】

以下是唯一的一种放置方案。



【样例 2 输入】

1

6

2

.....

3

.....

4

.....

5

.....

【样例 2 输出】

1

2003

【样例 3 输入】

1

7

2

IIIIIII

3

IIIIIII

4

IIIIIII

5

IIIIIII

【样例 3 输出】

1

5

【子任务】

保证 $1 \leq n \leq 2000$ 。

测试点编号	$n \leq$	特殊性质
1 ~ 2	1	无
3 ~ 9	7	
10 ~ 11	500	四行字符串完全相同，且仅由 “IOSZ” 组成
12 ~ 13	500	四行字符串完全相同，且仅由 “IOSTZ” 组成
14 ~ 15	500	四行字符串完全相同，且不包含 “.”
16 ~ 21	500	无
22 ~ 23	1000	
24 ~ 25	2000	