2025 年洛谷 SCP-J/S 能力认证

第二轮认证

提高级

第一试

时间：2025 年 10 月 18 日 14:30 ～ 18:30

题目名称	旅行	分割	列车	平局
题目类型	传统型	传统型	传统型	传统型
目录	`trip`	`divide`	`train`	`draw`
可执行文件名	`trip`	`divide`	`train`	`draw`
输入文件名	`trip.in`	`divide.in`	`train.in`	`draw.in`
输出文件名	`trip.out`	`divide.out`	`train.out`	`draw.out`
每个测试点时限	2.0 秒	1.0 秒	1.5 秒	2.0 秒
内存限制	512 MiB	512 MiB	512 MiB	1024 MiB
测试点数目	20	25	25	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	`trip.cpp`	`divide.cpp`	`train.cpp`	`draw.cpp`

编译选项

对于 C++ 语言	`-O2 -std=c++14`

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. 洛谷统一评测时采用的机器配置为：Intel(R) Xeon(TM) Platinum 8369HC CPU @3.30GHz，内存 24GB。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。

旅行（trip）

【题目背景】

积云厚重，而卷云飘渺。

【题目描述】

小 W 报名了一个为期 n 天的旅行团。作为一名气象学家，他记录了旅行期间每天的温度，形成一个序列 $A = (a_1, a_2, \ldots, a_n)$。

小 W 希望从这 n 天中选取一个**连续的时间段**进行研究。他的研究从第 l 天到第 r 天，其中 $1 \le l \le r \le n$。

对于一个选定的时间段，其温度序列为 $B = (a_l, a_{l+1}, \ldots, a_r)$。小 W 会计算这个序列 B 的**前缀和**序列 $C = (c_1, c_2, \ldots, c_k)$，其中 $k = r - l + 1$ 且 $c_i = \sum_{j=1}^{i} B_j$。

其中：$\sum_{j=1}^{i} B_j$ 即 $B_1 + B_2 + B_3 + \cdots + B_i$。

小 W 的任务是，在所有可能的连续时间段中，找出这样一个时间段，使其对应的前缀和序列 C 中**包含最多数量的** 0。请输出这个最大数量。

【输入格式】

从文件 ***trip.in*** 中读入数据。

在洛谷上提交程序时，无需从文件读取输入数据，也无需将答案输出到文件中。

本题包含多组测试数据。

第一行输入一个正整数 T，表示数据组数。

接下来包含 T 组数据，每组数据的格式如下：

- 第一行输入一个正整数 n。
- 第二行输入 n 个整数，表示温度序列 $a_1, a_2, \ldots, a_n$。

【输出格式】

输出到文件 ***trip.out*** 中。

在洛谷上提交程序时，无需从文件读取输入数据，也无需将答案输出到文件中。

对于每组测试数据：

- 输出一行一个非负整数，表示最优情况下前缀和序列中 0 的最大数量。

【样例 1 输入】

```
2
5
-1 0 1 0 0
5
4 2 0 -2 9
```

【样例 1 输出】

```
3
1
```

【样例 1 解释】

该样例共有 2 组测试数据。

对于第一组测试数据，温度序列为 $A = [-1, 0, 1, 0, 0]$。最佳选择是选取从第 1 天到第 5 天的时间段，对应的子序列为 $[-1, 0, 1, 0, 0]$。其前缀和序列计算如下：

- $c_1 = -1$
- $c_2 = -1 + 0 = -1$
- $c_3 = -1 + 0 + 1 = 0$
- $c_4 = -1 + 0 + 1 + 0 = 0$
- $c_5 = -1 + 0 + 1 + 0 + 0 = 0$

前缀和序列为 $[-1, -1, 0, 0, 0]$，其中包含 3 个 0。这是所有可能的时间段中能得到的最大数量，因此答案是 3。

对于第二组测试数据，温度序列为 $A = [4, 2, 0, -2, 9]$。最佳选择是选取从第 2 天到第 4 天的时间段，对应的子序列为 $[2, 0, -2]$。其前缀和序列计算如下：

- $c_1 = 2$
- $c_2 = 2 + 0 = 2$
- $c_3 = 2 + 0 + (-2) = 0$

前缀和序列为 $[2, 2, 0]$，其中包含 1 个 0。这是所有可能的时间段中能得到的最大数量，因此答案是 1。

【样例 2】

见选手目录下的 ***trip/trip2.in*** 与 ***trip/trip2.ans***。

【样例 2 解释】

该组样例共有 5 组测试数据。

其中第 i 组测试数据满足测试点 i 的限制，例如第 1 组测试数据满足测试点 1 的限制。下文同理。

【样例 3】

见选手目录下的 ***trip/trip3.in*** 与 ***trip/trip3.ans***。

【样例 3 解释】

该组样例共有 5 组测试数据。

其中第 i 组测试数据满足测试点 $i+5$ 的限制。

【样例 4】

见选手目录下的 ***trip/trip4.in*** 与 ***trip/trip4.ans***。

【样例 4 解释】

该组样例共有 5 组测试数据。

其中第 i 组测试数据满足测试点 $i+10$ 的限制。

【样例 5】

见选手目录下的 ***trip/trip5.in*** 与 ***trip/trip5.ans***。

【样例 5 解释】

该组样例共有 5 组测试数据。

其中第 i 组测试数据满足测试点 $i+15$ 的限制。

【子任务】

记 V 为所有 a_i 的绝对值的最大值，即 $\max\limits_{i=1}^{n}|a_i|$。

测试点	$n \leq$	$V \leq$	特殊性质
1, 2	10	10^6	无
$3 \sim 6$	500		
$7 \sim 10$	5,000		
$11 \sim 14$	10^5	1	
15, 16		10^6	A
17, 18			B
19			无
20	10^6		

- 特殊性质 A：保证 $n = 10^5$，且序列 A 随机生成。随机方式是在所有符合数据范围的序列 A 中，等概率均匀随机抽取得到输入时的序列 A。
- 特殊性质 B：保证对于每个 $i \in [1, n]$，$a_i \geq 0$。

对于所有测试数据，保证 $1 \leq T \leq 5$，$1 \leq n \leq 10^6$，$-10^6 \leq a_i \leq 10^6$。

分割（divide）

【题目描述】

你是洛咕咕王国的土地测绘官。洛咕咕王国并购了一块新的领土，这块新的领土正等待被分配。

这块领土可被认为是一棵有 n 个结点、结点编号为 1 到 n 的树，根为编号 1。为了便于表述，我们把每个结点 i 在原树中的深度记作 d_i，并规定根的深度为 1。

你的王国有若干位诸侯希望购买土地，因此现在要从这棵树中选出 k 个**两两不同**的结点，并把它们的编号排成一个有序序列 $b = (b_1, b_2, \ldots, b_k)$。这个序列必须满足两个条件：

第一，每个被选的结点都不是根，并且它们的深度是非降的，也就是对所有 $1 \le i < k$ 有 $1 < d_{b_i} \le d_{b_{i+1}}$。

第二，按照序列里每一个结点 b_i（$i = 1, 2, \cdots, k$），把它们各自与父亲的连边断开。断开这些 k 条边后，原树会被分成 $k+1$ 棵互不相交的连通子树。我们把这 $k+1$ 棵子树依次编号。其中，第 1 棵到第 k 棵对应于根为 $b_1, \ldots, b_k$ 的那 k 棵子树，而第 $k+1$ 棵子树则是剩下的、包含原来的树根 1 的那一棵（它的根仍记为 1）。对于第 i 棵子树，把该子树中所有结点在**原树**中的深度值去重后组成一个集合，记为 S_i。要求这次分割满足等式：

$$S_1 = \bigcap_{i=2}^{k+1} S_i$$

换言之，第 1 棵子树中出现的所有深度恰好是“出现在所有其他子树中的深度”的交集。

我们把任意两个序列 b 视为不同的方案当且仅当它们作为序列不同（即结点相同但顺序不同视为不同方案）。你的任务是计算满足上述条件的序列 b 的个数，对 998244353 取模后输出结果。

【输入格式】

从文件 ***divide.in*** 中读入数据。

在洛谷上提交程序时，无需从文件读取输入数据，也无需将答案输出到文件中。

第一行包含两个正整数 n, k，分别表示树的结点个数和需要选出的结点个数。

第二行包含 $n-1$ 个正整数，第 i 个正整数表示结点 $(i+1)$ 的父结点的编号 p_i。根结点 1 没有父结点。

【输出格式】

输出到文件 ***divide.out*** 中。

在洛谷上提交程序时，无需从文件读取输入数据，也无需将答案输出到文件中。

输出一行一个整数，表示满足题目条件的序列 b 的个数，结果对 998244353 取模。

【样例 1 输入】

```
11 2
1 2 3 1 1 5 6 8 1 10
```

【样例 1 输出】

```
4
```

【样例 1 解释】

如图，合法的序列 b 一共有 4 个，分别是：

- 令 $b_1 = 5, b_2 = 10$；
- 令 $b_1 = 10, b_2 = 5$；
- 令 $b_1 = 7, b_2 = 11$；
- 令 $b_1 = 11, b_2 = 7$。

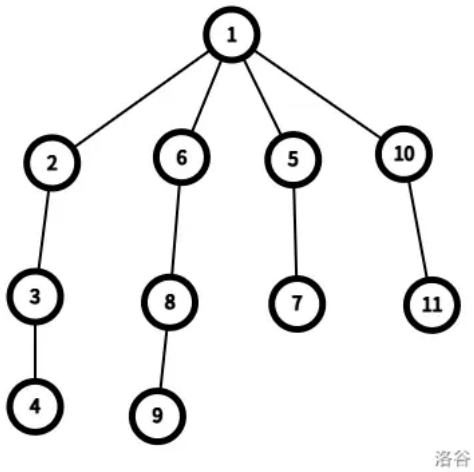

图 1: 样例 1

以 $b_1 = 5, b_2 = 10$ 为例，$d_5 = 2$ 且 $d_{10} = 2$，有 $d_5 \le d_{10}$。当我们切断结点 5 和 10 与其父结点 1 的连边后，原树被分割成三棵子树。第一棵子树以 $b_1 = 5$ 为根，包含结点 $\{5, 7\}$；第二棵子树以 $b_2 = 10$ 为根，包含结点 $\{10, 11\}$；第三棵子树则是包含原树根 1 的剩余部分。

对于第一棵子树，其结点在**原树**中的深度为 $\{2, 3\}$，因此 $S_1 = \{2, 3\}$。对于第二棵子树，其结点深度同样为 $\{2, 3\}$，所以 $S_2 = \{2, 3\}$。对于包含根结点的第三棵子树，去重后的深度集合为 $S_3 = \{1, 2, 3, 4\}$。计算交集可得 $S_2 \cap S_3 = \{2, 3\}$，与 S_1 相等。因此 $b_1 = 5, b_2 = 10$ 是一个符合条件的序列。

【样例 2 输入】

```
13 3
1 2 3 1 1 5 6 8 1 10 11 7
```

【样例 2 输出】

```
72
```

【样例 2 解释】

一个符合条件的序列 b 是 $b_1 = 4, b_2 = 9, b_3 = 12$。

【样例 3 输入】

```
7 3
1 1 1 1 2 3
```

【样例 3 输出】

```
12
```

【样例 4】

见选手目录下的 ***divide/divide4.in*** 与 ***divide/divide4.ans***。

【样例 4 解释】

这个样例满足测试点 8 的条件限制。

【样例 5】

见选手目录下的 ***divide/divide5.in*** 与 ***divide/divide5.ans***。

【样例 5 解释】

这个样例满足测试点 13 的条件限制。

【样例 6】

见选手目录下的 ***divide/divide6.in*** 与 ***divide/divide6.ans***。

【样例 6 解释】

这个样例满足测试点 21 ∼ 25 的条件限制。

【子任务】

测试点	$n \leq$	特殊性质
1 ∼ 2	12	无
3 ∼ 5	10^2	
6 ∼ 7	2,000	
8		A
9 ∼ 10	2×10^5	
11 ∼ 12	10^6	
13	2×10^5	B
14 ∼ 15	10^6	
16 ∼ 20	2×10^5	无
21 ∼ 25	10^6	

- 特殊性质 A：$k = 2$。
- 特殊性质 B：树为一棵满 t 叉树，其中 $t \in [3, n) \cap \mathbb{Z}$。

对于所有测试数据，保证 $2 \leq k < n \leq 10^6$，$1 \leq p_i < i$。树保证连通。

列车（train）

【题目描述】

洛咕咕王国是咕咕星球上领土最辽阔的王国，因此在洛咕咕王国中如何高效移动成为了一个难题。洛咕咕王国有 n 座城市，神奇的是这 n 座城市在一条直线上，我们把第 i 座城市在直线上的位置记作 p_i。这些城市是按照顺序进行编号的，因此 $p_1 < p_2 < \cdots < p_n$。

已知洛咕咕王国有一条按照顺序连接所有城市的铁路，洛咕咕王国在这条铁路上开行了多趟列车以满足各城市间的运输需求。起初对于每一对正整数对 (i, j)（$1 \le i < j \le n$），都有一趟以城市 i 为起点站、开往城市 j 且以城市 j 为终点站的列车。除开起点站和终点站外，这趟列车还会依次停靠中途城市 $i+1, i+2, \ldots, j-1$。

为了减轻洛咕咕王国票务系统的压力，洛咕咕王国开行的所有列车的收费标准实行一票制。洛咕咕咕民只要乘坐从以城市 i 为起点站、城市 j 为终点站的列车，票价均为起点与终点位置之差 $p_j - p_i$，与洛咕咕咕民实际乘坐的区间无关。

洛咕咕王国接下来会按照顺序发生 m 次事件，第 i 次事件为以下两种类型之一：

1. 洛咕咕国王命令停开所有起点站城市编号大于等于 x_i 且终点站编号小于等于 y_i 的列车。一旦一趟列车在某次事件中被停开，它在后续所有时刻都视作已停开，不会被恢复。
2. 一位洛咕咕咕民查询搭乘一趟未停开的列车从城市 x_i 搭乘至城市 y_i 的最小花费，若不存在这样的列车则输出 `-1`。若一趟列车先后停靠城市 l 和城市 r 两个站点，则称可以搭乘这趟列车从城市 l 到城市 r，一趟列车的起点站和终点站也算入这趟列车的停靠范围。

【输入格式】

从文件 ***train.in*** 中读入数据。

在洛谷上提交程序时，无需从文件读取输入数据，也无需将答案输出到文件中。

本题包含多组测试数据。

第一行包含一个正整数 T，表示数据组数。

接下来包含 T 组数据，每组数据的格式如下：

- 第一行包含一个正整数 n, m，表示城市个数。
- 第二行包含 n 个正整数 $p_1, p_2, \ldots, p_n$，表示各个城市在直线上的位置。
- 接下来 m 行每行表示一次事件的发生。首先读入一个正整数 o 表示事件类型：
  - 若 $o = 1$ 表示发生了一个类型 1 事件，接下来两个正整数 x_i, y_i 表示停开所有起点站城市编号大于等于 x_i 且终点站编号小于等于 y_i 的列车。
  - 若 $o = 2$ 表示发生了一个类型 2 事件，接下来两个正整数 x_i, y_i 表示查询搭乘一趟未停开列车从城市 x_i 搭乘至城市 y_i 的最小花费。

【输出格式】

输出到文件 ***train.out*** 中。

在洛谷上提交程序时，无需从文件读取输入数据，也无需将答案输出到文件中。

对于每组数据：输出若干行，对于每个类型 2 事件输出一行一个整数表示答案：若存在对应的列车可以搭乘则输出最小花费，否则输出 `-1`。

【样例 1 输入】

```
2
4 6
1 2 3 4
2 1 3
2 3 4
1 2 3
2 2 3
1 1 4
2 1 4
5 5
1 4 5 7 1000000000
1 2 4
2 3 5
2 2 3
1 1 2
2 3 4
```

【样例 1 输出】

```
2
1
2
-1
999999995
4
6
```

【样例 1 解释】

在第一组测试数据中，最初共有 6 趟列车开行。

- 第 1 个事件：查询从城市 1 搭乘至城市 3 的最小花费。当前所有列车均在开行，因此最优的方案是搭乘以城市 1 为起点站、城市 3 为终点站的列车，花费为 $p_3 - p_1 = 3 - 1 = 2$。
- 第 2 个事件：查询从城市 3 搭乘至城市 4 的最小花费。当前所有列车均在开行，因此最优的方案是搭乘以城市 3 为起点站、城市 4 为终点站的列车，花费为 $p_4 - p_3 = 4 - 3 = 1$。
- 第 3 个事件：停开所有起点站城市编号大于等于 2，终点站城市编号小于等于 3 的列车，即停开以城市 2 为起点站、城市 3 为终点站的列车。
- 第 4 个事件：查询从城市 2 搭乘至城市 3 的最小花费。由于以城市 2 为起点站、城市 3 为终点站的列车已被停开，所以**无法搭乘这趟列车**。最优方案之一是搭乘以城市 1 为起点站、城市 3 为终点站的列车，花费为 $p_3 - p_1 = 3 - 1 = 2$，可以证明不存在花费更小的方案。
- 第 5 个事件：停开所有起点站城市编号大于等于 1，终点站城市编号小于等于 4 的列车，即停开所有列车。以城市 2 为起点站、城市 3 为终点站的列车先前已被停开，本次事件将不会对这趟列车产生任何影响。
- 第 6 个事件：查询从城市 1 搭乘至城市 4 的最小花费。由于所有列车已被停开，无法从城市 1 搭乘至城市 4 ，故输出 `-1`。

对于第二组测试数据，我有一个绝佳的解释，但是这里空间太小写不下。

【样例 2】

见选手目录下的 ***train/train2.in*** 与 ***train/train2.ans***。

【样例 2 解释】

该组样例满足测试点 1 的限制。

【样例 3】

见选手目录下的 ***train/train3.in*** 与 ***train/train3.ans***。

【样例 3 解释】

该组样例满足测试点 4 的限制。

【样例 4】

见选手目录下的 ***train/train4.in*** 与 ***train/train4.ans***。

【样例 4 解释】

该组样例满足测试点 8 的限制。

【样例 5】

见选手目录下的 ***train/train5.in*** 与 ***train/train5.ans***。

【样例 5 解释】

该组样例满足测试点 11 的限制。

【样例 6】

见选手目录下的 ***train/train6.in*** 与 ***train/train6.ans***。

【样例 6 解释】

该组样例满足测试点 13 的限制。

【样例 7】

见选手目录下的 ***train/train7.in*** 与 ***train/train7.ans***。

【样例 7 解释】

该组样例满足测试点 15 的限制。

【样例 8】

见选手目录下的 ***train/train8.in*** 与 ***train/train8.ans***。

【样例 8 解释】

该组样例满足测试点 19 的限制。

【样例 9】

见选手目录下的 ***train/train9.in*** 与 ***train/train9.ans***。

【样例 9 解释】

该组样例满足测试点 23 的限制。

【子任务】

测试点	$n \leq$	特殊性质
$1 \sim 3$	10^2	无
$4 \sim 7$	3,000	
$8 \sim 10$	5×10^4	A
11, 12		B
13, 14		C
$15 \sim 18$		D
$19 \sim 22$		E
$23 \sim 25$	10^5	无

若第 i 次事件为类型 1，则令 S_i 为第 i 次事件所有被停开的列车（不一定是本次事件后才被停开的列车）所构成的集合。

- 特殊性质 A：保证不存在正整数 i, j 满足 $1 \leq i < j \leq m$ 且第 i 次事件为类型 2，第 j 次事件为类型 1。
- 特殊性质 B：保证不存在正整数 i, j 满足 $1 \leq i < j \leq m$ 且第 i 次事件和第 j 次事件均为类型 1 且 $S_i \cap S_j \neq \varnothing$。
- 特殊性质 C：保证不存在正整数 i, j 满足 $1 \leq i < j \leq m$ 且第 i 次事件和第 j 次事件均为类型 1 且 $S_i \not\subseteq S_j$。
- 特殊性质 D：对于每次事件，均保证 x_i, y_i 在所有可能的 x_i, y_i 中等概率选取。
- 特殊性质 E：保证 $p_n = n$。

对于所有测试数据，保证：$1 \leq T \leq 10$，$2 \leq n, m \leq 10^5$，$1 \leq x < y \leq n$，$1 \leq p_1 < p_2 < \cdots < p_n \leq 10^9$。

平局（draw）

【题目描述】

有 n 个人站成一排玩石头剪刀布，每个人只会出一种固定的手势。

定义 **最大平局问题** 为以下问题：

定义一次操作为：

- 选择任意的相邻两人，让他们玩石头剪刀布，将败者移除。若平局，则任意移除一人。然后，其余的人会补上空位。

已知当双方手势不同时，石头可以战胜剪刀，剪刀可以战胜布，布可以战胜石头；若一局中双方手势相同，则该局游戏为平局。

你需要不断执行操作直到只剩下一个人。最大化操作造成的平局次数。

现在这 n 个人的手势还未确定。你需要求出所有满足限制的确定手势的方案的 **最大平局问题** 的答案之和，对 10^9+7 取模。

具体的，限制为一个长度为 n 的正整数序列 a，且满足 $1 \le a_i \le 7$，a_i 的含义为：

- 当 $a_i \in \{1,3,5,7\}$ 时，可以将第 i 个人的手势确定为剪刀。
- 当 $a_i \in \{2,3,6,7\}$ 时，可以将第 i 个人的手势确定为石头。
- 当 $a_i \in \{4,5,6,7\}$ 时，可以将第 i 个人的手势确定为布。

【输入格式】

从文件 ***draw.in*** 中读入数据。

在洛谷上提交程序时，无需从文件读取输入数据，也无需将答案输出到文件中。

第一行一个整数 n。

接下来输入一个长度为 n 的字符串，其中第 i 个字符代表 a_i。

【输出格式】

输出到文件 ***draw.out*** 中。

在洛谷上提交程序时，无需从文件读取输入数据，也无需将答案输出到文件中。

一行一个整数，表示答案对 10^9+7 取模的结果。

【样例 1 输入】

```
6
421234
```

【样例 1 输出】

```
5
```

【样例 1 解释】

下文用 $\underline{SRP}$ 来分别表示剪刀、石头、布。

对于第一组数据，满足限制的确定手势的方案共有两种，分别为 $\underline{PRSRSP}$ 和 $\underline{PRSRRP}$。

对于前者，一种最大化平局次数的操作方案是 $\underline{PRSRSP} \to \underline{PRRSP} \to \underline{PRRP} \to \underline{PRP} \to \underline{PP} \to \underline{P}$，共出现了两次平局。

对于后者，一种最大化平局次数的操作方案是 $\underline{PRSRRP} \to \underline{PRRRP} \to \underline{PRRP} \to \underline{PRP} \to \underline{PP} \to \underline{P}$，共出现了三次平局。

所以答案是 $2+3=5$。

【样例 2 输入】

```
7
2111473
```

【样例 2 输出】

```
23
```

【样例 3 输入】

```
15
266165141645216
```

【样例 3 输出】

```
906
```

【样例 4 输入】

```
25
7772717273647537773772342
```

【样例 4 输出】

```
477398784
```

【样例 5】

见选手目录下的 ***draw/draw5.in*** 与 ***draw/draw5.ans***。

【样例 5 解释】

该样例满足测试点 7 ∼ 9 的约束条件。

【样例 6】

见选手目录下的 ***draw/draw6.in*** 与 ***draw/draw6.ans***。

【样例 6 解释】

该样例满足测试点 10 ∼ 11 的约束条件。

【样例 7】

见选手目录下的 ***draw/draw7.in*** 与 ***draw/draw7.ans***。

【样例 7 解释】

该样例满足测试点 12 ∼ 13 的约束条件。

【样例 8】

见选手目录下的 ***draw/draw8.in*** 与 ***draw/draw8.ans***。

【样例 8 解释】

该样例满足测试点 16 ∼ 19 的约束条件。

【样例 9】

见选手目录下的 ***draw/draw9.in*** 与 ***draw/draw9.ans***。

【样例 9 解释】

该样例满足测试点 $22 \sim 25$ 的约束条件。

【子任务】

测试点	$n \leq$	特殊性质
$1 \sim 2$	10	无
$3 \sim 6$	15	
$7 \sim 9$	3,000	$a_i \in \{1, 2, 3\}$
$10 \sim 11$	50	$a_i = 7$
$12 \sim 13$		无
$14 \sim 15$	400	$a_i = 7$
$16 \sim 19$		无
$20 \sim 21$	3,000	$a_i = 7$
$22 \sim 25$		无

对于所有测试数据，保证：$1 \leq n \leq 3000$，$1 \leq a_i \leq 7$。