

NOIP 2025 模拟赛

ZHZX Round 2

一、题目概况

中文题目名	集合	挑战自动机	排列	干掉理性龙
英文题目名	set	dfa	perm	dragon
源程序	set.cpp	dfa.cpp	perm.cpp	dragon.cpp
输入文件	set.in	dfa.in	perm.in	dragon.in
输出文件	set.out	dfa.out	perm.out	dragon.out
题目类型	传统题	传统题	传统题	传统题
时间限制	2s	2s	1s	2.5 ~ 7s
空间限制	1024MB	256MB	512MB	128MB
子任务数目	8	8	8	12
子任务依赖	是	是	是	是
结果比较方式	全文比较（过滤行末空格及文末回车）			

二、注意事项

1. 使用 `-std=c++14 -O2 -static` 进行编译。
2. 评测环境为 NOI Linux 2.0。
3. 评测机比较 low，请注意代码常数。
4. 禁止使用 `pragma` 和 `attribute`。

1 集合 (set)

1.1 题目描述

有 T 次询问。每次给定 n ，求合法集合的个数。

定义一个集合 S 合法，当且仅当：

- $S \subseteq \{x : x \mid n\}, S \neq \emptyset, \min(S) \geq 2$.
- $\forall x, y \in S (x \neq y), \gcd(x, y) = 1$.

请输出答案对 998244353 取模后的结果。

1.2 输入格式

第一行，一个正整数 T ，表示询问次数。

接下来 T 行，每行一个正整数 n 。

1.3 输出格式

共 T 行，第 k 行表示第 k 次询问的答案。

1.4 样例一

见下发文件中的 `ex_set1.in` 和 `ex_set1.out`。

在第一组数据中， $n = 2$ ，只有集合 $\{2\}$ 满足条件。

在第二组数据中， $n = 4$ ，只有集合 $\{2\}$ 与 $\{4\}$ 满足条件。但是集合 $\{2, 4\}$ 不满足条件，因为 $\gcd(2, 4) \neq 1$ ；集合 $\{3\}$ 也不满足条件，因为 $3 \nmid n$ 。

在第三组数据中， $n = 6$ ，只有集合 $\{2\}, \{3\}, \{6\}, \{2, 3\}$ 满足条件。

1.5 样例二

见下发文件中的 `ex_set2.in` 和 `ex_set2.out`。

1.6 样例三

见下发文件中的 `ex_set3.in` 和 `ex_set3.out`。

1.7 样例四

见下发文件中的 `ex_set4.in` 和 `ex_set4.out`，该样例满足 $T = 10^3, n \leq 10^8$ ，且前 500 组数据满足 $\omega(n) = 1$ ，后 500 组数据满足 $\omega(n) \leq 2$ 。

1.8 样例五

见下发文件中的 `ex_set5.in` 和 `ex_set5.out`, 该样例满足 $T = 10^3, n \leq 10^6$ 。

1.9 样例六

见下发文件中的 `ex_set6.in` 和 `ex_set6.out`, 该样例满足 $T = 10^4, n \leq 10^7$ 。

1.10 样例七

见下发文件中的 `ex_set7.in` 和 `ex_set7.out`, 该样例满足 $T = 10^4, n \leq 10^8$ 。

1.11 限制与约定

对于 100% 的数据, 保证 $1 \leq T \leq 10^6, 1 \leq n \leq 10^8$ 。

可能需要更快的输入输出方式。

下表中 $\omega(n)$ 表示 n 的本质不同质因子个数。

子任务编号	$T \leq$	$n \leq$	特殊性质	分值	子任务依赖
1	10	10	无	10	无
2	100	100		10	1
3	10^6	10^8	$\omega(n) = 1$	10	无
4			$\omega(n) \leq 2$	10	3
5	10^3	10^6	无	10	2
6	10^6			10	5
7		10^7		20	6
8		10^8		20	4, 7

2 挑战自动机 (dfa)

2.1 题目描述

Fer 珍藏着一台确定性有限状态自动机，字符集为 $\{i, f\}$ ，共有 n 个状态，状态 u 的颜色为 $c_u \in \{0, 1\}$ ，初始状态为 a 。

出于某些原因，这台自动机没有终止状态，这是无关紧要的。

如果你不了解自动机理论，可以简单理解为这是一个有向图，每个点恰好有两条出边，分别标记为 $u(i), u(f)$ 。

转移某个字符，即选择该字符指示的出边。

转移某个字符串，即依次转移每个字符。

注意，如果理解为有向图，可能存在重边和自环。

Fer 并不知道，卷毛怪偷偷复刻了这台自动机，除了初始状态不同（记作 b ），其余结构完全一致。

由于自动机的内部结构过于复杂，作为普通人类的你能只能观测当前状态的颜色。

定义一个字符串合法（能够证明两台自动机不同），当且仅当，两台自动机分别转移该字符串后，颜色不同。

你想知道，合法字符串的最小长度是多少？如果不存在合法字符串，输出 `Impossible`。

2.2 输入格式

多测，首先输入一个正整数 T ，表示接下来有 T 组数据。

每组数据形如：

第一行，一个正整数 n ，表示状态数。

第二行， n 个用空格分开的正整数，第 k 个表示状态 k 通过字符 i 转移后的状态。

第三行， n 个用空格分开的正整数，第 k 个表示状态 k 通过字符 f 转移后的状态。

第四行， n 个用空格分开的整数，表示序列 c 。

第五行，两个正整数 a 和 b ，分别表示两台自动机的初始状态。

2.3 输出格式

共 T 行，第 k 行表示第 k 组数据的答案。

如果有解，输出合法字符串的最小长度；否则输出 `Impossible`。

2.4 样例一

见下发文件中的 `ex_dfa1.in` 和 `ex_dfa1.out`。

在第一组数据中，一种可能的合法字符串是 `iff`，此时达到最小长度。

转移过程为 $(1, 2) \rightarrow (2, 3) \rightarrow (3, 4) \rightarrow (4, 5) \rightarrow (5, 6)$ ，此时两个状态颜色不同。

容易说明答案不可能比 4 更小。

在第二组数据中，注意到所有转移方式都不会改变状态，显然无解。

2.5 样例二

见下发文件中的 `ex_dfa2.in` 和 `ex_dfa2.out`。

2.6 样例三

见下发文件中的 `ex_dfa3.in` 和 `ex_dfa3.out`。

2.7 样例四

见下发文件中的 `ex_dfa4.in` 和 `ex_dfa4.out`。

2.8 限制与约定

对于 100% 的数据，保证 $1 \leq T \leq 500, n \geq 2, \sum n \leq 2 \times 10^6, a \neq b, c_a = c_b$ 。

可以说明，如果有解，答案不会超过 10^{18} 。

极限输入量超过 32 MB，可能需要更快的输入方式。

保证 std 不使用更快的输入方式。

子任务编号	$\sum n \leq$	特殊性质	分值	子任务依赖
1	5×10^3	无	10	无
2	5×10^4		10	1
3	2×10^5	$\sum c_u \leq 1$	15	无
4		通过字符 <code>i</code> 转移时，状态不变	5	无
5		无	10	2 ~ 4
6	2×10^6	$\sum c_u \leq 1$	15	3
7		通过字符 <code>i</code> 转移时，状态不变	5	4
8		无	30	5 ~ 7

3 排列 (perm)

3.1 题目描述

给定整数序列 a ，长度为 n ，下标从 1 开始。

定义 n 阶排列为形如 $p: [n] \rightarrow [n]$ 的双射，令 \mathfrak{S}_n 为所有 n 阶排列构成的集合。

给定正整数 C ，你需要计算

$$\sum_{\pi \in \mathfrak{S}_n} \prod_{i=1}^{n-1} [a_{\pi(i)} a_{\pi(i+1)} \leq C]$$

其中记号 $[P]$ 表示命题 P 的真值。

请输出答案对 $(10^9 + 7)$ 取模后的结果。

3.2 输入格式

第一行，两个正整数 n 和 C ，分别表示序列长度和乘积限制。

第二行， n 个用空格分开的整数，表示序列 a 。

3.3 输出格式

只有一行，表示答案。

3.4 样例一

见下发文件中的 `ex_perm1.in` 和 `ex_perm1.out`，该样例满足子任务 1 的限制。

3.5 样例二

见下发文件中的 `ex_perm2.in` 和 `ex_perm2.out`，该样例满足子任务 3 的限制。

3.6 样例三

见下发文件中的 `ex_perm3.in` 和 `ex_perm3.out`，该样例满足子任务 4 的限制。

3.7 样例四

见下发文件中的 `ex_perm4.in` 和 `ex_perm4.out`，该样例满足子任务 8 的限制。

3.8 限制与约定

对于 100% 的数据，保证 $1 \leq n \leq 5 \times 10^3, 1 \leq |a_i|, C \leq 10^9$ 。

注意区分整数与正整数。

子任务编号	$n \leq$	特殊性质	分值	子任务依赖
1	10	无	4	无
2	16		16	1
3	50		16	2
4	200	$a_i \geq 1$	12	无
5		无	12	3, 4
6	5×10^3	$a_i \geq 1$	16	4
7		$ a_i \leq 2$	8	无
8		无	16	5 ~ 7

4 干掉理性龙 (dragon)

不要滥用评测，建议特判一定会 T 的子任务。

4.1 题目背景

既然十字神明都能选入 NOIP 联考，那就无所谓了。

这是一道半原创题，如果您不屑于做，至少可以欣赏一下。

4.2 题目描述

给定序列 a ，长度为 n ，下标从 1 开始。给定阈值 V ，保证 $a_i \in [0, V) \cap \mathbb{N}$ 。

有 m 次操作，每次操作为下列类型之一：

1. 给定 $[l, r] \subseteq [1, n], x \in [0, V) \cap \mathbb{N}$ ，执行 $\forall i \in [l, r] \cap \mathbb{N}, a_i \leftarrow x$ 。
2. 给定 $[l, r] \subseteq [1, n], x \in [0, V) \cap \mathbb{N}$ ，执行 $\forall i \in [l, r] \cap \mathbb{N}, a_i \leftarrow (a_i + x) \bmod V$ 。
3. 给定 $[l, r] \subseteq [1, n], x \in [0, V) \cap \mathbb{N}$ ，询问是否 $\exists i, j \in [l, r] \cap \mathbb{N}, \text{s.t. } a_i - a_j = x$ 。
4. 给定 $[l, r] \subseteq [1, n], x \in [0, V) \cap \mathbb{N}$ ，询问是否 $\exists i, j \in [l, r] \cap \mathbb{N}, \text{s.t. } a_i + a_j = x$ 。
5. 给定 $[l, r] \subseteq [1, n], x \in [0, V) \cap \mathbb{N}$ ，询问是否 $\exists i, j \in [l, r] \cap \mathbb{N}, \text{s.t. } a_i \times a_j = x$ 。

注意询问操作中的 i, j 可以相等。注意只有操作 2 在模意义下进行。

对于每个询问操作，输出 **Yes** 或者 **No**，**强制在线**。

4.3 输入格式

第一行，四个正整数 n, m, V, id ，分别表示序列长度、操作次数、阈值和子任务编号（样例为 0）。

第二行， n 个用空格分开的非负整数，表示序列 a 。

接下来 m 行，每行四个正整数，形如 `opt l r x`，表示一次操作（`opt` 表示操作类型）。

注意，本题强制在线。

具体而言，假设给出的区间是 $[l, r]$ ，则真实的区间为 $[\min(l \oplus g, r \oplus g), \max(l \oplus g, r \oplus g)]$ 。

其中 \oplus 表示按位异或运算， g 为上一个答案为 **Yes** 的询问的真实区间左端点（如果不存在则为 0）。

4.4 输出格式

对于每个操作 3, 4, 5，输出一行 **Yes** 或者 **No**。

4.5 样例一

见下发文件中的 `ex_dragon1.in` 和 `ex_dragon1.out`。

4.6 样例二

见下发文件中的 `ex_dragon2.in` 和 `ex_dragon2.out`，该样例满足子任务 2 的限制。

4.7 样例三

见下发文件中的 `ex_dragon3.in` 和 `ex_dragon3.out`，该样例满足子任务 3 的限制。

4.8 样例四

见下发文件中的 `ex_dragon4.in` 和 `ex_dragon4.out`，该样例满足子任务 4 的限制。

4.9 样例五

见下发文件中的 `ex_dragon5.in` 和 `ex_dragon5.out`，该样例满足子任务 5 的限制。

4.10 样例六

见下发文件中的 `ex_dragon6.in` 和 `ex_dragon6.out`，该样例满足子任务 6 的限制。

4.11 样例七

见下发文件中的 `ex_dragon7.in` 和 `ex_dragon7.out`，该样例满足子任务 8 的限制。

4.12 样例八

见下发文件中的 `ex_dragon8.in` 和 `ex_dragon8.out`，该样例满足子任务 12 的限制。

4.13 限制与约定

对于 100% 的数据, 保证 $1 \leq n, m, V \leq 1.5 \times 10^5$, V 是 64 的倍数, 解密后 $1 \leq l \leq r \leq n$, 操作 2 不超过 5×10^4 次。

可能需要更快的输入输出方式。

注意不同寻常的时空限制。

对于子任务 1 ~ 7, 时间限制为 2.5s。

对于子任务 8, 时间限制为 4s。

对于子任务 9 ~ 11, 时间限制为 5.5s。

对于子任务 12, 时间限制为 7s。

保证 std 不需要特判任何子任务。

子任务编号	$n \leq$	$m \leq$	$V \leq$	特殊性质	分值	子任务依赖	
1	10^3	10^3	10^3	无	4	无	
2	10^4	10^4	10^4		16	1	
3	10^5	10^5	64		20	无	
4	1.5×10^5	1.5×10^5	1.5×10^5	解密后 $l = 1, r = n$	8	无	
5				解密后 $l = 1$, 没有修改操作	8	无	
6				没有修改操作	12	5	
7				没有操作 2, 序列和操作随机	12	无	
8	10^5	10^5	10^5	无	4	2, 3	
9	1.5×10^5				4	8	
10	10^5	1.5×10^5	1.5×10^5		4	8	
11		10^5			4	8	
12	1.5×10^5	1.5×10^5			4	4, 6, 7, 9 ~ 11	