

OI Round2 Div2.5

OI Contest

时间: █ 年 █ 月 █ 日 █:█ ~ █:█

题目名称	小木棍	献祭	染色体	线段树二分
题目类型	传统型	传统型	传统型	传统型
目录	<code>polygon</code>	<code>xor</code>	<code>lx</code>	<code>segbs</code>
可执行文件名	<code>polygon</code>	<code>xor</code>	<code>lx</code>	<code>segbs</code>
输入文件名	<code>polygon.in</code>	<code>xor.in</code>	<code>lx.in</code>	<code>segbs.in</code>
输出文件名	<code>polygon.out</code>	<code>xor.out</code>	<code>lx.out</code>	<code>segbs.out</code>
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	2.0 秒
内存限制	256 MiB	256 MiB	256 MiB	256 MiB
预估	██████	██████	██████	██████
测试点数目	20	5	5	60
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	<code>ltm.cpp</code>	<code>xor.cpp</code>	<code>lx.cpp</code>	<code>segbs.cpp</code>
-----------	----------------------	----------------------	---------------------	------------------------

编译选项

对于 C++ 语言	<code>-O2 -std=c++14 -static -m32</code>
-----------	--

注意事项（请仔细阅读）

1. 本场比赛为阶段学习评估，总体难度较小，与 ██████████-Div2.5 难度相近。
2. 由于评测机性能原因，评测时题目时间限制可能会进行调整。
3. 时间限制、空间限制分别不小于标准程序运行时间和内存使用的 1.1 倍、1.1 倍。
4. 本场比赛可支持 C++14 语言。
5. 本场比赛不提供 ██████████。
6. 本校模拟赛题不得外传，包括但不限于：私自拷题，在任何网站记录题目信息（包括洛谷非公开云剪贴板，非公开题目等），向在线 AI 询问题目，或 ██████████。
7. 题目背景为架空世界，请勿将现实中人物对号入座，否则 ██████████。
8. 300pts+ 并且猜出题目中的歌曲者可向出题人领赏 30RMB。
9. 请把 █████ 读作【数据删除】。
10. █████ 造数据，欢迎各种乱搞，各位选手 AK 后请勿大声喧哗。

小木棍 (polygon)

【题目背景】

九点一刻。

盯着刚刚通过的 polygon 大样例，十分的无聊。

为了让自己能安心睡着，给自己出了一道 polygon-Pro-Max-Ultra 版。

由于他真的盯着 polygon-Pro-Max-Ultra 睡着了，所以问题就留给你了。

【题目描述】

给你 n 根小木棍，长度 a_1, a_2, \dots, a_n 。你需要选出一个可重集 s ，规则如下：

1. 将 a 分成一些可重集 x_1, x_2, \dots, x_k 使 a 中任意元素都恰好属于某一个可重集。
2. 选择 x_i 的众数加入 s ($i \in [1, k]$)。

现在问你不同的 s 有多少种，答案对 $(114514 + 1919810)$ 取模。

【输入格式】

从文件 *polygon.in* 中读入数据。

第一行一个整数 n ，表示木棍个数。

第二行 n 个整数，第 i 个整数表示 a_i 。

【输出格式】

输出到文件 *polygon.out* 中。

一行一个正整数，表示答案。

【样例 1 输入】

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

【样例 1 输出】

```
1 4
```

【样例 1 解释】

所有可能的 s 有如下四种情况： $\{2\}, \{2, 2\}, \{1, 2\}, \{1, 2, 2\}$

【样例 2 输入】

```
1 10  
2 1 1 1 2 2 2 3 3 3 4
```

【样例 2 输出】

```
1 126
```

【样例 3】

见选手目录下的 *polygon/polygon3.in* 与 *polygon/polygon3.ans*, 该样例满足 $n = 5000$ 。

【数据范围】

测试点编号	特殊性质	n
1	无	5
2	A	50
3		
4	无	500
5		
6	A	5000
7		
8 ~ 20	无	

对于 100% 的测试点，保证 $n \leq 5000, a_i \leq n$ ($i \in [1, n]$)

献祭 (xor)

【题目背景】

本题难度预估：下位绿。

我知道你肯定是做第一题看不下去了才跑过来看第二题的。

你现在如果再去看一眼第三题，你可能会丧失更多对这套题的兴趣。

但是这道题真的是一道水题。

而且第四题比这道题还要水。

就算你依然不会，你也应该认真思考，如何找到题目的突破口。

这对于思维训练的帮助是巨大的。

这也是我们打模拟赛的根本原因。

【题目描述】

异或之神的面前，出现了 n 个献祭者。

每个献祭者体内都用一种神秘物质：异或之液。

每个献祭者都有一个编号以及异或之液的纯度，编号从 1 到 n 连续。

这些献祭者按照编号从小到大被排成一排。

共有 m 次献祭仪式，每一次献祭仪式，异或之神都会选择若干名编号连续的献祭者，其中编号最小的为 l ，编号最大的为 r ，且 $r - l > 1$ 。

然后，他会将选出的这些献祭者按照编号从小到大抽取他们的异或之液，并按顺序注入一条可以玻璃管内。

注意，不同献祭者的异或之液目前是不会相融的。

注入完之后会把玻璃管切成若干段，注意，他只会在两种纯度不同的异或之液的交界处进行切割。

切割后，玻璃管被分成 k 段，并且 $k \geq 2$ 。

紧接着，异或之神就会使用魔法将每一段内的异或之液开始融合。

融合后的异或之液的纯度为融合前所有异或之液纯度的异或和。

当每一段中的异或之液都充分融合后，异或之神开始检查每一段的纯度。

如果所有段的纯度都相等，那么异或之神就会得到异或之液的法力，从而提升自己的异或水平，否则他将不会有任何变化。

现在，他想知道，对于给定的 l, r 存不存在一种切割方案使得他可以提升自己的异或水平。

如果存在输出 poly，否则输出 modui。

【输入格式】

从文件 **xor.in** 中读入数据。

第一行包含两个整数 n 和 m 。

下一行包含 n 个整数 a_1, \dots, a_n 。

接下来的每行 q 包含两个整数 l 和 r ，分别描述查询。

【输出格式】

输出到文件 *xor.out* 中。

对于每个查询，输出”poly”或”modui”。

【样例 1 输入】

```
1          4
2          5 5
3          1 1 2 3 0
4          1 5
5          2 4
6          3 5
7          1 3
8          3 4
9          5 5
10         1 2 3 4 5
11         1 5
12         2 4
13         3 5
14         1 3
15         2 3
16         7 4
17         12 9 10 9 10 11 9
18         1 5
19         1 7
20         2 6
21         2 7
22         11 4
23         0 0 1 0 0 1 0 1 1 0 1
24         1 2
25         2 5
26         6 9
27         7 11
```

【样例 1 输出】

```
1          YES
2          YES
3          NO
4          NO
5          NO
6
7          YES
8          NO
9          NO
10         YES
11         NO
12
13        NO
14        NO
15        NO
16        NO
17
18        YES
19        NO
20        YES
21        YES
```

【样例 1 解释】

本题没有多组数据，方便排版将 4 组样例压在同一样例中。

【数据范围】

对于 20% 的数据， $n, m \leq 100$ 。

对于 100% 的数据， $n, m \leq 2 \times 10^5$ 。

染色体 (lx)

【题目背景】

蒹葭苍苍，白露为霜。
所谓伊人，在水一方。
溯洄从之，道阻且长。
溯游从之，宛在水中央。
蒹葭萋萋，白露未晞。
所谓伊人，在水之湄。
溯洄从之，道阻且跻。
溯游从之，宛在水中坻。
蒹葭采采，白露未已。
所谓伊人，在水之涘。
溯洄从之，道阻且右。
溯游从之，宛在水中沚。

【题目描述】

本题难度预估：下位紫。

本题有多组数据。

白露再次凝成霜，一年又一年，来自 ZJ 的 LX 大佬却不会长大，因为他有着像李渊一样的心智，不过嘛，终究还是要来到初中。

踏入新的省份，新的班级，LX 大佬是多么的陌生，望着周围新的环境、新的同学、新的老师以及那新的关系，他有些迷茫，但他那李渊一样的心智帮了他，让他渐渐地融入了这个新的班级。

不过，LX 大佬引起了有个同学 Fish 的注意，因为 LX 是那么的活泼，和谁都能玩在一起，又是那么的优秀，常年霸榜逆序铜牌，Fish 不知怎的，似乎有种莫名其妙的感觉在她的内心荡漾。

渐渐地，他们也熟识了，从陌生人到交往，从交往到熟识，从熟识到朋友，从朋友到……

随着时间的推移，LX 和 Fish 也开始了信件上的单独相约，从生快的祝福，到绚丽的诗文，他享受着 Fish 信件的陪伴。

可是 Fish 不知道的是，LX 大佬从出生起，就比常人多了一条 21 号染色体。

基因的缺陷始终令 LX 自卑，不敢直面自己，看到作为正常人的 Fish 愉悦的活着的时候，他总感到难以自拔的悲伤。

明智的 LX 大佬意识到，他必须学会拒绝 Fish。

当 LX 大佬向 Fish 坦白一切时，Fish 彻底崩溃。

从此以后，Fish 的信件中遍充斥着血泪的控诉与呼喊，可 LX 仍然不为所动，再未回过一封。

直到某一天，一封信引起了 LX 的注意。

这封信写得工工整整，恰好写满了 n 行， n 列的作文纸。

待他仔细看向信件的内容时，却看作文纸的每个格子中，都是一个数字。

学过 MO 的 LX 大佬一眼看出了 Fish 的把戏。

第 i 行，第 j 列的数中就是 $f(\gcd(i, j))$ 。

$f(i)$ 是斐波那契数列的第 i 项。

Fish 在信件的背面写道：

To LX:

所有格子中数字的乘积，模 998244353，就是你所期待的，安史之乱的密码。祝好。

高傲的 LX 大佬却热泪盈眶。

他不敢相信，Fish 竟然为了他，不知付出多少努力，找到了消去他那可恶的染色体的关键数字。可惜，计算那 n^2 个数的乘积实在是过于庞大，他难以计算。

所以他找到了你。

形式化题意：

给你一个 $n \times n$ 的方阵 A ， $A_{i,j} = \gcd(f(i), f(j))$ ，请你求出 A 中所有数的乘积模 998244353 的值。

【输入格式】

从文件 *lx.in* 中读入数据。

【输出格式】

输出到文件 *lx.out* 中。

Output.

【样例 1 输入】

1

Sample Input.

【样例 1 输出】

1

Sample Output.

【样例 1 解释】

Note.

【样例 2】

见选手目录下的 *lx/lx2.in* 与 *lx/lx2.ans*。

【数据范围】

对于 60% 的数据， $n, m \leq 10^5$ 。

对于 100% 的数据， $n, m \leq 3 \times 10^7$ 。

线段树二分 (segbs)

【题目背景】

题目名称与本题做法没有关联。

某场模拟赛正在进行。

Uit: T1 怎么做啊?

(T1 正解是莫队)

Col: 用树剖写。

十五分钟后。

Uit: 树剖过了。

【题目描述】

本题难度预估：上位黄。

给你一个长度为 n 的序列 a , m 次操作。

操作 1: 给定 l, r , 问 a_l, a_{l+1}, \dots, a_r 是否两两不同。

操作 2: 给定 l, r , 问 a_l, a_{l+1}, \dots, a_r 是否两两相同。

操作 3: 给定 l, r , 问 a_l, a_{l+1}, \dots, a_r 是否是一个从 1 到 $r - l + 1$ 的排列。

操作 4: 给定 l, r, s, t , 问可重集 $\{a_l, a_{l+1}, \dots, a_r\}$, $\{a_s, a_{s+1}, \dots, a_t\}$ 是否完全相同。

顾名思义，可重集就是可以有相同元素的集合。

【输入格式】

从文件 *segbs.in* 中读入数据。

【输出格式】

输出到文件 *segbs.out* 中。

Output.

【样例 1 输入】

1

Sample Input.

1

Sample Output.

【样例 1 解释】

Note.

【样例 2】

见选手目录下的 *segbs/segbs2.in* 与 *segbs/segbs2.ans*。

【数据范围】

对于 25% 的数据， $n, m \leq 10^3$ 。

对于 50% 的数据， $n, m \leq 10^5$ 。

对于 75% 的数据， $n, m \leq 10^6$ 。

对于 100% 的数据， $n, m \leq 3 \times 10^7$ 。

对于每一档不同的 n, m 范围，都有 15 个点，分别对应 4 个操作的所有组合。