

# 冲刺NOIP2022模拟试题

时间：7:40-12:10

(请选手务必仔细阅读本页内容)

## 一. 题目概况

中文题目名称	黎曼几何	代数几何	基因进化	几何考试
英文题目与子目录名	riemannian	algebraic	reverse	exams
可执行文件名	riemannian	algebraic	reverse	exams
输入文件名	riemannian.in	algebraic.in	reverse.in	exams.in
输出文件名	riemannian.out	algebraic.out	reverse.out	exams.out
每个测试点时限	1.0秒	1.0秒	1.0秒	1.0秒
测试点数目	10	10	10	10
测试点是否等分	是	是	是	是
附加样例文件	有	有	有	有
结果比较方式	全文比较	全文比较	全文比较	全文比较
题目类型	传统型	传统型	传统型	传统型
运行内存上限	512MB	512MB	512MB	512MB

## 二. 提交源程序文件名

对于C++语言	riemannian.cpp	algebraic.cpp	reverse.cpp	exams.cpp
---------	----------------	---------------	-------------	-----------

## 三. 编译选项

对于C++语言	-lm -std=c++11 -O2
---------	--------------------

## 四. 注意事项

- 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、全国统一评测时采用的机器配置为：Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz，内存 8GB。上述时限以此配置为准。
- 4、只提供 Linux 格式附加样例文件。
- 5、特别提醒：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其为准。

# 黎曼几何 ( riemannian )

## 【问题描述】

在乏味的黎曼几何课上，你为了打发时间，玩起了硬币。

你在桌子上画了三个圈，这三个圈顺时针编号为 1, 2, 3。首先，你将  $N$  枚硬币全部垒放在 1 号圈里面。这些硬币从下到上面值依次为  $N, N - 1, \dots, 1$ 。你每次将某一个圈里面值最小的硬币移动到顺时针相邻的下一个圈里 (即  $1 \rightarrow 2, 2 \rightarrow 3, 3 \rightarrow 1$ )。并且，你需要保证，目标圈里的所有硬币都比你移动的这个硬币的面值大。这样的一次操作称为“一步”。

为了不让老师察觉到你的行为，你决定尽快将这些硬币全部移动到某一个格子。由于你还没有想好移动到哪个格子可以更快完成，你决定先计算出移动到 2 号格子格子需要的最少步数。

(实际上，根据数据范围，你在移动总重量为  $10^6$  吨的硬币)

## 【输入格式】

输入文件名为 *riemannian.in*。

第一行一个正整数  $T$ ，代表数据组数。

接下来共  $T$  行，每一行一个正整数  $N$ ，代表硬币的总数。

## 【输出格式】

输出文件名为 *riemannian.out*。

假设，对于第  $i (i \in [1, T])$  组数据，将所有硬币移动到第 2 个圈的最小步数为  $a_i$ ，移动到第 3 个圈上的最小步数为  $b_i$ 。那么，最终你的答案文件中只需要出现一行，用空格隔开的两个数字，分别为  $(a_1 \bmod P) \oplus (a_2 \bmod P) \oplus \dots \oplus (a_T \bmod P)$ ，和  $(b_1 \bmod P) \oplus (b_2 \bmod P) \oplus \dots \oplus (b_T \bmod P)$ 。其中  $\oplus$  代表答案的按位异或操作， $P = 998244353$ ， $\bmod$  为取模操作。

## 【样例输入1】

```
3
1
2
3
```

## 【样例输出1】

```
11 16
```

## 【样例1说明】

三次询问的  $a$  分别为 1, 5, 15， $b$  分别为 2, 7, 21。

## 【样例2】

见选手目录下的 *riemannian/riemannian2.in* 与 *riemannian/riemannian2.ans*。

### 【数据范围及约定】

测试点编号	$n$ 的范围	$T$ 的范围
1	$n \leq 3$	$T \leq 10^3$
2, 3	$n \leq 11$	$T \leq 10^3$
4	$n \leq 15$	$T \leq 10^3$
5	$n \leq 100$	$T \leq 10^3$
6, 7	$n \leq 10^5$	$T \leq 10^3$
8, 9	$n \leq 10^{12}$	$T \leq 10^5$
10	$n \leq 10^{12}$	$T \leq 1.2 \times 10^6$

对于全部的测试点,  $n \leq 10^{12}, T \leq 1.2 \times 10^6$ 。

# 代数几何 ( algebraic )

## 【问题描述】

我们将代数和几何结合在一起，就有了代数几何 (*algebraic geometry*)。

现在，我们有两个一模一样的环，每个环上有  $n$  个数字。如果将环从中间任意一个地方剪开，则每一个环都是一个长度为  $n$  的序列  $a_0 \dots a_{n-1}$ 。初始，这两个环是重合的，对应位置的数字完全相同。

现在，我们将其中一个环旋转一个角度，使得这个环与另一个环对应的数字恰好岔开了  $k$  个。也就是，现在这个环的第  $i$  个数字与另一个环的第  $(i + k) \bmod n$  对应。现在，我们求这两个环对应位置的数相乘，然后把乘积加起来，得到  $S$ 。

你的任务是将  $a$  重新排列，使得  $S$  尽可能大。输出这个最大的  $S$ 。

## 【输入格式】

输入文件为 *algebraic.in*。

第一行一个正整数  $n$ ，代表环的长度。

第二行  $n$  个正整数  $a_i$ ，代表一开始环上的数字。

第三行一个正整数  $m$ ，代表给定的不同的  $k$  的个数。

第四行开始共  $m$  行，每行一个正整数  $k_i$ 。

## 【输出格式】

输出文件为 *algebraic.out*。

对于每个  $k$ ，均输出一行一个正整数，表示最大的  $S$ 。

## 【样例输入1】

```
6
1 2 3 4 5 6
3
1
2
0
```

## 【样例输出1】

```
82
85
91
```

## 【样例说明1】

当  $k = 1$  时，一种可能的排列方式是：1, 2, 4, 6, 5, 3。还有一种可能的排列方式是：1, 3, 5, 6, 4, 2。

## 【样例2】

见选手目录下的 *algebraic/algebraic2.in* 与 *algebraic/algebraic2.ans*。

## 【数据范围及约定】

测试点编号	$n$ 的范围	$m$ 的范围	$k$ 的范围
1	$n \leq 10$	$m \leq 10$	$0 \leq k < n$
2, 3	$n \leq 18$	$m \leq 18$	$0 \leq k < n$
4	$n \leq 36$ 且为偶数	$m = 1$	$k = 2$
5, 6	$n \leq 1000$	$m \leq 10$	$k = 1$
7	$n \leq 50$	$m \leq 10$	$0 \leq k \leq 2$
8, 9	$n \leq 1000$	$m \leq 10$	$0 \leq k < n$
10	$n \leq 5000$	$m \leq n$	$0 \leq k < n$

对于全部的测试点， $2 \leq n \leq 5000$ ,  $1 \leq m \leq n$ ,  $0 \leq k < n$ ,  $1 \leq a_i \leq 10^5$ 。

# 基因进化 ( reverse )

## 【问题描述】

2049年，一种基因修改技术在  $Y$  大陆流行；

为了简化问题，我们把基因看做一个长度为  $N$  的正整数序列；

人们可以对基因进行多次修改。具体来讲，对于每次修改可以表述为：选择这个基因序列的一个前缀，然后翻转它们；

但是这个技术有很多限制，首先，必须先翻转长度小的前缀，再翻转长度大的前缀；

其次，有一些长度的前缀是无法被翻转的，这样的前缀一共有  $M$  个；

大部分人都认为，字典序越小的基因序列越优越，越先进，所以它们想知道，在通过这个技术翻转基因序列后，所能得到的最小字典序的序列是什么。

## 【输入格式】

从文件 *reverse.in* 中读取数据。

一行一个整数  $op$ ，描述测试数据组数。

对于每组测试数据：

第一行两个整数  $N, M$ ，描述基因序列的长度，和无法翻转的前缀数量；

接下来一行  $N$  个正整数  $a_1 \dots a_n$ ，描述基因序列；

接下来一行  $M$  个不同的数  $b_1 \dots b_m$ ，描述不能被翻转的前缀的数量；

## 【输出格式】

输出到文件 *reverse.out* 中。

输出  $op$  行，输出能获得的的最小字典序序列经过哈希后的结果，哈希一个基因序列  $s_1 \dots s_n$  的方法如下：

$$HashCode(s) = (\sum_{i=1}^n 37^{i-1} * s_i) \mod 998244353$$

## 【样例输入1】

```
5
4 0
2 2 1 2
5 0
1 1 1 1 1
3 0
3 2 1
3 1
3 2 1
3
3 2
3 2 1
3 2
```

### 【样例输出1】

```
104119
1926221
4182
1482
1446
```

### 【样例解释1】

5 组数据最后的基因序列的样子分别是：1, 2, 2, 2, 1, 1, 1, 1, 1, 1, 2, 3, 2, 3, 1, 3, 2, 1。

### 【样例2】

见选手目录下的 *reverse/reverse2.in* 与 *reverse/reverse2.ans*。

### 【数据范围及约定】

对于 20% 的数据： $N \leq 20$ ；

对于 40% 的数据： $N \leq 2000$ ；

对于 60% 的数据： $N \leq 50000$ ；

对于另外 20% 的数据： $M = 0$

对于 100% 的数据， $0 \leq M \leq N \leq 300000$ ， $T \leq 100$ ， $\sum N \leq 1000000$ ， $1 \leq a_i \leq 10^9$ ， $1 \leq b_i \leq N$ 。

# 几何考试 ( exams )

## 【问题描述】

由于你在黎曼几何课堂上玩弄硬币，在代数几何课堂上玩剪纸，到考试的时候你发现你看不懂试卷了。

于是，你打算预测一下其他同学的排名来打发考试时间。

在考试结束之前，所有学生的成绩都是不确定的。但是由于每个学生的水平存在差距，成绩并不会完全随机，你还是有机会预测一个学生的排名的。

学生  $i$  的总分将会是区间  $[L_i, R_i]$  内均匀分布的一个实数。

现在，请你求出每个同学的期望排名。排名的定义是考试结束后，分数大于等于自己分数的人的个数。即，最高分的排名为 1，以此类推。如果你的输出与标准答案的相对误差不超过  $10^{-6}$  即算正确。

## 【输入格式】

输入文件名为 *exams.in*。

第一行一个整数  $N$ ，代表学生的总数。

接下来  $N$  行，每行两个整数  $L_i, R_i$ ，代表学生总分的分布区间。

## 【输出格式】

输出文件名为 *exams.out*。

每行一个实数，代表学生的期望排名。建议保留 7 位或更多位小数。

## 【样例输入1】

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

## 【样例输出1】

```
2.83333333
1.16666667
2.00000000
```

## 【样例说明1】

第一位同学永远要排在第二位同学的后面，但是有  $1/6$  的概率得分超过第三位同学，因此期望排名为  $1 + 5/6 \approx 2.83333$ 。

同理，第二位同学有  $5/6$  的概率得分超过第三位同学，因此期望排名为  $1 + 1/6 \approx 1.16667$ 。

不难验证，第三位同学的期望排名是 2.00000。



## 【样例2】

见选手目录下的 *exams/exams2.in* 与 *exams/exams2.ans*。

## 【数据范围及约定】

测试点编号	$N$ 的范围	$L_i, R_i$ 的范围
1, 2	$N \leq 3$	$L_i, R_i \leq 5$
3, 4	$N \leq 10$	$L_i, R_i \leq 100$
5, 6, 7	$N \leq 1000$	$L_i, R_i \leq 10^5$
8, 9	$N \leq 10^5$	$L_i, R_i \leq 10^5, R_i - L_i$ 为定值
10	$N \leq 10^5$	$L_i, R_i \leq 10^5$

对于全部的测试点， $1 \leq N \leq 10^5, 1 \leq L_i < R_i \leq 10^5$ 。