

# 2025 年联赛层基础能力水平测试

## Round 2

时间: 2025 年 4 月 4 日 07:50 ~ 12:00

题目名称	序列加法机	摸鱼军训	皮卡丘	银行的源起
题目类型	传统型	传统型	传统型	传统型
目录	seq	bubble	pi	banking
可执行文件名	seq	bubble	pi	banking
输入文件名	seq.in	bubble.in	pi.in	banking.in
输出文件名	seq.out	bubble.out	pi.out	banking.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	256 MB	512 MB	256 MB	256 MB
测试点数目	3	10	20	5
测试点是否等分	否	是	是	否

提交源程序文件名

对于 C++ 语言	seq.cpp	bubble.cpp	pi.cpp	banking.cpp
-----------	---------	------------	--------	-------------

编译选项

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

### 注意事项与提醒 (请选手务必仔细阅读)

1. 选手请直接提交源程序至 [becoder.com.cn](http://becoder.com.cn) 上的对应比赛。
2. 输入输出文件名必须使用英文小写。
3. 选手提交的源程序必须存放在**已建立好的，且带有样例文件和下发文件**的文件夹中，文件夹名称与对应试题英文名一致。
4. 文件名 (包括程序名和输入输出文件名) 必须使用英文小写。
5. C++ 中函数 `main()` 的返回值类型必须是 `int`，值必须为 0。
6. 若无特殊说明，结果比较方式为**忽略行末空格、文末回车后的全文比较**。
7. 程序可使用的栈空间大小与该题内存空间限制一致。
8. 在终端中执行命令 `ulimit -s unlimited` 可将当前终端下的栈空间限制放大，但你使用的栈空间大小不应超过题目限制。
9. 每道题目所提交的**代码文件大小限制为 100KB**。
10. 若无特殊说明，输入文件与输出文件中同一行的相邻整数均使用一个空格分隔。

11. 输入文件中可能存在行末空格，请选手使用更完善的读入方式（例如 `scanf` 函数）避免出错。

12. 直接复制 PDF 题面中的多行样例，数据将带有行号，建议选手直接使用对应目录下的样例文件进行测试。

13. 使用 `std::deque` 等 STL 容器时，请注意其内存空间消耗。

14. 请务必使用题面中规定的的编译参数，保证你的程序在本机能够通过编译。此外不允许在程序中手动开启其他编译选项，一经发现，本题成绩以 0 分处理。15. 对于因未遵守以上规则对成绩造成的影响，相关申诉不予受理。

## 序列加法机 (seq)

### 【题目描述】

A 酱发明了序列加法机，它的作用是把一个长度为  $n$  的单调不降序列  $a$  转换成另一个长度为  $n$  的单调不降序列  $b$ 。不过由于序列加法机的功能还不完善，现在它只能执行不超过  $m$  次以下操作：

- 选择一个  $1 \leq i \leq n$  和一个整数  $x$ ，将  $a_i$  加上  $x$  ( $x$  可以  $< 0$ )，操作代价是  $x^2$

并且序列加法机还需要保证每次操作完成后， $a$  依然是单调不降的，那么序列加法机最少需要多少代价才能将  $a$  转换成  $b$ ？

### 【输入格式】

从文件 `seq.in` 中读入数据。

第一行，两个正整数  $n, m$ 。

接下来两行，每行  $n$  个数，分别表示  $a$  和  $b$ 。

### 【输出格式】

输出到文件 `seq.out` 中。

一个数，表示最小代价，对 998244353 取模。

无解输出 -1。

### 【样例 1 输入】

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

### 【样例 1 输出】

```
1 7
```

### 【样例 2】

见附加文件中的 `seq/ex_seq2.in` 与 `seq/ex_seq2.ans`。

### 【数据范围】

对于所有数据，保证  $1 \leq n, m \leq 10^5, 0 \leq a_i, b_i \leq 10^9$ ，保证  $\{a\}, \{b\}$  两个序列单调不降。

子任务编号	分值	特殊性质
1	10	$n \leq 3, m, \sum  a_i - b_i  \leq 10$
2	40	$n, m \leq 300$
3	50	无特殊限制

## 摸鱼军训 (bubble)

### 【题目描述】

冒泡演出是 CS 大学的军训结束表演，由  $n$  位同学组成，每位同学都领取一个从 1 到  $n$  的编号，同学们的编号两两不同。

冒泡演出的内容很简单，刚开始  $n$  位同学站成一排，同学们的编号形成一个排列  $a[1 \dots n]$ ，教官每吹一次哨，同学们就进行一次冒泡排序：

```
1 for (int i = 1; i < n; i++) {  
2     if (a[i] > a[i + 1]) swap(a[i], a[i + 1]);  
3 }
```

众所周知，教官至多吹  $n - 1$  次哨后，同学们手中的编号恰好就会形成升序排列  $1, 2, \dots, n$ ，不过因为同学们都想摸鱼，所以  $n$  位同学一共提出了  $q$  个以下格式的询问：

- $(k, x)$ ：询问持有编号  $x$  的同学在教官吹  $k$  次哨后所在位置的下标。

### 【输入格式】

从文件 *bubble.in* 中读入数据。

第一行 1 个整数  $n$ ，表示参加冒泡演出的同学的数量。

第二行  $n$  个整数  $a[1 \dots n]$ ，保证是一个排列，表示同学们的初始排列。

第三行 1 个整数  $q$ ，表示同学们的询问总数。

接下来  $q$  行，每行 2 个整数  $(k, x)$  代表一次询问。

### 【输出格式】

输出到文件 *bubble.out* 中。

输出  $q$  行，每行 1 个整数代表答案。

### 【样例 1 输入】

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

### 【样例 1 输出】

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

**【样例 2 输入】**

```
1 5
2 4 5 2 3 1
3 5
4 3 1
5 4 2
6 3 3
7 2 4
8 1 5
```

**【样例 2 输出】**

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

**【样例 3】**

见附加文件中的 bubble/ex\_bubble3.in 与 bubble/ex\_bubble3.ans。

**【数据范围】**

对于 20% 的数据,  $n, q \leq 2000$ 。

对于另 20% 的数据,  $n, q \leq 10^5$ , 不同的  $k$  取值不超过 20 种。

对于另 20% 的数据,  $n, q \leq 10^5$ , 不同的  $x$  取值不超过 20 种。

对于另 20% 的数据,  $n, q \leq 10^5$ 。

对于 100% 的数据,  $1 \leq n, q \leq 5 \times 10^5, 1 \leq k < n, 1 \leq x \leq n, a[1 \dots n]$  保证是一个排列。

注意输入输出量较大, 请使用快速的读写方式。

## 皮卡丘 (pi)

### 【题目描述】

小 D 的皮卡丘可以产生  $n$  种不同的电压，第  $i$  种电压值为  $a_i$ 。当皮卡丘需要释放电击技能时，需要选择两种电压  $l, r (l \leq r)$  对应产生的电压值为  $a_l - a_r$ 。

小 D 对他的皮卡丘进行了  $m$  次训练，训练有以下两种：

- 增幅训练，小 D 按照指标  $l, r, x$  训练皮卡丘，使得皮卡丘能释放的第  $l$  种电压到第  $r$  种电压的电压值  $+=x$ 。
- 攻击训练，小 D 按照指标  $l, r, k$  训练皮卡丘，皮卡丘在区间  $[l, r]$  中选出电压值并打出  $k$  次电击。为了提升训练效果，这  $k$  次电击的组合方式必须两两不相同（不同攻击训练使用的电击技能之间没有限制）。两次电击的组合方式相同的定义是对于它们所使用的电压值对  $(a_{l_0}, a_{r_0}), (a_{l_1}, a_{r_1})$  有  $l_0 = l_1$  且  $r_0 = r_1$ 。

小 D 想知道，在每次攻击训练时，皮卡丘最大能打出多少电击伤害？电击伤害定义为训练中所有电击技能电压值之和。

### 【输入格式】

从文件 *pi.in* 中读入数据。

第一行两个整数  $n, m$  意义如题面所示。

接下来一行  $n$  个整数，表示皮卡丘能释放的电压值。

接下来  $m$  行每行表示一次训练：

- 增幅训练形如  $1\ l\ r\ x$ 。
- 攻击训练形如  $2\ l\ r\ k$ 。

### 【输出格式】

输出到文件 *pi.out* 中。

对于每次攻击训练，输出一行表示皮卡丘在这次训练中最高可以打出多少伤害。

### 【样例 1 输入】

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

**【样例 1 输出】**

```

1 5
2 7
3 2

```

**【样例 1 解释】**

第一次攻击训练，皮卡丘可以释放电击  $(a_4, a_4), (a_4, a_5), (a_4, a_6)$ ，产生的电击伤害为

$$(5 - 5) + (5 - 1) + (5 - 4) = 5$$

第二次攻击训练，皮卡丘可以释放电击  $(a_3, a_5), (a_4, a_5)$ ，产生的电击伤害为

$$(4 - 1) + (5 - 1) = 7$$

第三次攻击训练，此时电压序列变为了  $[-1, -1, 2, 4, 2, 5]$ ，皮卡丘可以释放电击  $(a_2, a_2), (a_3, a_3), (a_3, a_5), (a_4, a_4), (a_4, a_5)$ ，产生的电击伤害为

$$(-1 - (-1)) + (2 - 2) + (2 - 2) + (4 - 4) + (4 - 2) = 2$$

**【样例 2】**

见附加文件中的 pi/ex\_pi2.in 与 pi/ex\_pi2.ans。

**【数据范围】**

对于测试点 1-2，满足  $n, m \leq 100$

对于测试点 3-7，满足  $n, m \leq 1000, \sum k \leq 2000$

对于测试点 8-10，满足  $n, m \leq 1000$

对于测试点 11-13，满足  $k = 1$

对于测试点 14-16，满足没有修改操作

对于测试点 17-20，没有特殊限制

对于所有的数据， $n, m \leq 10^5, 1 \leq l \leq r \leq n, \sum k \leq 3 \times 10^5, -10^6 \leq a_i, x \leq 10^6$

保证每次攻击训练一定能打出  $k$  次互不完全相同的电击。

为了完成训练指标，皮卡丘有时不得不打出电压为负数的电击。



## 银行的源起 (banking)

### 【题目描述】

Tree 国有  $T$  个省，每个省的省内道路组成一棵 Tree。这天 Tree 国的国王得到了神明的赐福：在 Tree 国开设“银行”吧！Tree 国的商人们会在“银行”的帮助下给你带来源源不尽的金银财宝！不过目前 Tree 国国库不是很富裕，Tree 国只能在每个省开设两家银行。

既然如此，那就最大化银行的性价比吧！假设 Tree 国某个省有  $n$  座城市，城市之间的道路当然满足 Tree 的约束，通行某条道路需要花费相应的时间。该省第  $i$  个城市生活着  $a_i$  个居民，当某位居民需要银行的服务时，该居民自然会去离自己城市最近的银行。可想而知，在未来该省两家银行开业的那一天，该省所有居民都会去离家最近的银行办卡，因此该省银行的性价比就定义为该省所有居民去往银行需要花费的时间之和。花费时间和越少性价比越高。

那么对于每个省，最优的银行建设方案是什么呢？

### 【输入格式】

从文件 *banking.in* 中读入数据。

第一行一个正整数  $T$ ，表示 Tree 国的省份数量。

对于每个省：

- 第一行一个正整数  $n$ ，表示该省有  $n$  个城镇。
- 第二行  $n$  个正整数  $a_1, a_2, \dots, a_n$ ，表示每个城镇的居民数。
- 接下来  $n - 1$  行，每行三个正整数  $u, v, w$ ，表示城镇  $u$  和城镇  $v$  之间有一条道路，通行该道路需要花费  $w$  的时间。

### 【输出格式】

输出到文件 *banking.out* 中。

对于每个省，输出一行一个正整数，表示该省最优的银行建设方案中，所有居民去往银行需要花费的时间之和。

### 【样例 1 输入】

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

7	5	1	2
8	6	1	1
9	7	2	1
10	8	1	2
11	9	1	1
12	10	2	2
13	10		
14	2	1	10 1 3 6 10 10 3 9
15	2	1	2
16	3	1	3
17	4	1	3
18	5	1	4
19	6	1	2
20	7	1	3
21	8	3	3
22	9	2	2
23	10	3	3

【样例 1 输出】

1	59
2	128

【样例 2】

见附加文件中的 banking/ex\_banking2.in 与 banking/ex\_banking2.ans。

【数据范围】

对于所有数据，保证  $2 \leq n \leq 10^5, 1 \leq a_i, w \leq 5 \times 10^4, 1 \leq u, v \leq n, 2 \leq \sum n \leq 5 \times 10^5$ 。

子任务编号	分值	特殊性质
1	10	$n, m \leq 10$
2	10	特殊性质 A
3	20	$2 \leq n \leq 10^3, \sum n \leq 2.5 \times 10^5$
4	20	$2 \leq n \leq 5 \times 10^4, \sum n \leq 2.5 \times 10^5$
5	40	无

特殊性质 A：保证树的形态是一条链。