

# 必可赛前公益众筹赛

## 第十一试

时间: 2024 年 11 月 26 日 07:50 ~ 12:15

题目名称	珠子	机器人	树	钻石岛
题目类型	传统型	传统型	传统型	传统型
目录	bead	robot	tree	island
可执行文件名	bead	robot	tree	island
输入文件名	bead.in	robot.in	tree.in	island.in
输出文件名	bead.out	robot.out	tree.out	island.out
每个测试点时限	1.0 秒	2.0 秒	1.0 秒	1.0 秒
内存限制	128 MB	512 MB	512 MB	512 MB
测试点数目	10	25	20	6
测试点是否等分	是	是	是	否

提交源程序文件名

对于 C++ 语言	bead.cpp	robot.cpp	tree.cpp	island.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 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。

## 珠子 (bead)

### 【题目描述】

有  $n$  颗珠子，珠子的编号依次为 1 到  $n$ 。第  $i$  颗珠子的美丽值为  $v_i$ ，这是一个在 1 到  $m$  之间的正整数。

小 Z 将会从中等概率随机抽取 1 到  $n$  颗珠子，那么此时他的高兴度为所有抽到的珠子的美丽值的最大公约数。小 Z 想知道，他的期望高兴度是多少，你能帮帮他吗？因为小 Z 不喜欢实数和大数，所以你只需要输出结果  $\times(2^n-1)$  的值对 998244353 取模的结果即可。

### 【输入格式】

从文件 *bead.in* 中读入数据。

第一行，包含两个正整数  $n$  和  $m$ 。

第二行，包含  $n$  个正整数  $v_1, v_2, \dots, v_n$ 。

### 【输出格式】

输出到文件 *bead.out* 中。

输出一行一个整数，即答案。

### 【样例 1 输入】

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

### 【样例 1 输出】

```
1 14
```

### 【数据范围】

对于 30% 的数据， $n \leq 20, m \leq 100000$ 。

对于另外 30% 的数据， $n \leq 100000, m \leq 20$ 。

对于 100% 的数据， $n \leq 100000, m \leq 100000$ 。

## 机器人 (robot)

### 【题目描述】

小 A 有一条数轴，于是他邀请小 B 一起来玩游戏。

小 A 和小 B 各有一个机器人。0 时刻时，他们都处于数轴的整点上（并不一定处于相同的位置）。

小 A 会向他的机器人依次下达  $n$  条指令（指令编号从 1 开始）。其中第  $i$  条指令的内容是：在接下来的  $A_i$  个时刻都以  $V_i$  每时刻的速度在数轴上**匀速运动**（ $V_i$  的取值范围为  $\{-1, 0, 1\}$ ）。这意味着，小 A 的机器人会在  $\left[\sum_{j=1}^{i-1} A_j, \sum_{j=1}^i A_j\right)$  时刻内，每时刻都向正方向移动  $V_i$  单位长度。

小 B 也会类似地向他的机器人依次下达  $m$  条指令。第  $i$  条指令的内容是：在接下来的  $B_i$  个时刻都以  $W_i$  每时刻的速度在数轴上**匀速运动**（ $W_i$  的取值范围同样为  $\{-1, 0, 1\}$ ）。

由于小 A 和小 B 将会同时结束游戏，所以有  $\sum_{i=1}^n A_i = \sum_{i=1}^m B_i$ ，我们也认为这个时刻是整个游戏的最后一个时刻，我们将它记作  $L$ 。

对于任意的**整数**时刻，如果两个机器人处于坐标轴上相同位置，那么小 A 会认为这个时刻是**优秀的**。

由于游戏时间太长，小 A 和小 B 早就忘记了他们机器人的初始坐标。不过他们都把自己的所有指令按顺序记录了下来。

小 B 对优秀的时刻数非常感兴趣。他想知道，对于所有可能的两机器人起始位置情况，最多可能有多少优秀的时刻。

无助的小 B 并不能弄清机器人的运行过程，所以请你帮帮他。**注意：从每一个区域出发的答案独立。**

### 【输入格式】

从文件 `robot.in` 中读入数据。

本题包含多组数据。第一行一个非负整数  $T$  表示数据组数，接下来依次描述每组数据。对于每组数据：

第 1 行一个正整数  $n$ ，表示小 A 的指令数。

第 2 行到第  $n+1$  行，第  $i+1$  行两个用空格隔开的整数  $V_i, A_i$ ，描述小 A 的第  $i$  条指令。

第  $n+2$  行一个正整数  $m$ ，表示小 B 的指令数。

第  $n+3$  行到第  $n+m+2$  行，第  $i+n+2$  行两个用空格隔开的整数  $W_i, B_i$ ，描述小 B 的第  $i$  条指令。

### 【输出格式】

输出到文件 `robot.out` 中。

对于每组数据，输出一行一个整数表示答案。

**【样例 1 输入】**

```
1 2
2 1
3 1 2
4 2
5 1 1
6 -1 1
7 1
8 0 6
9 4
10 -1 2
11 1 1
12 -1 2
13 1 1
```

**【样例 1 输出】**

```
1 2
2 3
```

**【样例 1 解释】**

对于第一组数据，小 A 和小 B 的机器人的初始坐标如果均为 0，即会有 2 个优秀的时刻：

时刻 0 两机器人的坐标分别为 0 和 0，是优秀的时刻。

时刻 1 两机器人的坐标分别为 1 和 1，是优秀的时刻。

时刻 2 两机器人的坐标分别为 2 和 0。

可以证明不可能存在更多优秀的时刻。

对于第二组数据，小 A 和小 B 的机器人的初始坐标如果分别为 2 和 4，即会有 3 个优秀的时刻（时刻 2, 4, 6）。可以证明不可能存在更多优秀的时刻。

**【样例 2】**

见附加文件中的 robot/ex\_robot2.in 与 robot/ex\_robot2.ans。

**【数据范围】**

对于一个测试点， $\sum(n + m)$  指的是该测试点中所有数据  $n + m$  的总和。

对于所有测试点，保证  $\sum(n + m) \leq 2 \times 10^6$ 。

对于所有测试点的所有数据, 保证  $1 \leq n, m \leq 10^5$ ,  $V_i, W_i \in \{-1, 0, 1\}$ ,  $1 \leq A_i, B_i \leq L \leq 10^{18}$ 。

测试点编号	$n, m$	$L$	$V_i$	$W_i$	$\sum(n + m)$
1	$\leq 100$	$\leq 200$	$\in \{-1, 0, 1\}$	$\in \{-1, 0, 1\}$	$\leq 1000$
2	$\leq 1000$	$= n = m$	$\in \{-1, 0, 1\}$	$\in \{-1, 0, 1\}$	$\leq 10000$
3 ~ 4	$\leq 1000$	$\leq 3000$	$\in \{-1, 0, 1\}$	$\in \{-1, 0, 1\}$	$\leq 10000$
5 ~ 6	$\leq 1000$	$\leq 10^9$	$\in \{-1, 0, 1\}$	$\in \{-1, 0, 1\}$	$\leq 10000$
7 ~ 8	$\leq 10^5$	$= n = m$	$\in \{-1, 0, 1\}$	$\in \{-1, 0, 1\}$	$\leq 10^6$
9 ~ 10	$\leq 10^5$	$2 \times 10^5$	$\in \{-1, 0, 1\}$	$\in \{-1, 0, 1\}$	$\leq 2 \times 10^6$
11	$\leq 10^5$	$10^{18}$	$\in \{0\}$	$\in \{-1, 0, 1\}$	$\leq 2 \times 10^6$
12 ~ 14	$\leq 10^5$	$10^{18}$	$\in \{0, 1\}$	$\in \{-1, 0\}$	$\leq 2 \times 10^6$
15 ~ 18	$\leq 10^5$	$10^{18}$	$\in \{0, 1\}$	$\in \{0, 1\}$	$\leq 2 \times 10^6$
19 ~ 25	$\leq 10^5$	$10^{18}$	$\in \{-1, 0, 1\}$	$\in \{-1, 0, 1\}$	$\leq 2 \times 10^6$

## 树 (tree)

### 【题目描述】

小 Z 给你一个以 1 为根的  $n$  个节点的有根树，其点集为  $V$ ，每个点有点权  $a_i$ 。

定义一个点集  $V' \subseteq V$  是好的，当且仅当存在  $V'$  中的一点  $p$ ，使得  $V'$  中任两个不同点的 lca 均为  $p$ 。

现在小 Z 想要把点集  $V$  不重不漏地划分成若干个好的点集。对于一个划分方案  $S$ ，其权值定义为：

$$\prod_{V' \in S} \left( \sum_{v \in V'} a_v \right)$$

求所有划分方案的权值和，答案对  $m$  取模。

### 【输入格式】

从文件 `tree.in` 中读入数据。

第一行两个正整数  $n, m$  表示树所包含的点数及模数。

第二行  $n$  个非负整数  $a_i$  表示每个点的点权。

接下来  $n - 1$  行，每行两个正整数  $u, v$ ，表示树上的一条边  $(u, v)$ 。保证给出的所有边与点构成一棵树。

### 【输出格式】

输出到文件 `tree.out` 中。

一行一个正整数，表示所有划分方案的权值和，对  $m$  取模。

### 【样例 1 输入】

```
1 3 998244353
2 1 2 3
3 1 2
4 1 3
```

### 【样例 1 输出】

```
1 29
```

### 【样例 1 解释】

共有 4 个划分方案： $\{\{1\}, \{2\}, \{3\}\}$ ， $\{\{1, 2\}, \{3\}\}$ ， $\{\{1, 3\}, \{2\}\}$ ， $\{\{1, 2, 3\}\}$ ，权值分别为 6, 9, 8, 6，和为 29。

**【样例 2 输入】**

```
1 15 20210416
2 2041996 16316965 19194939 4894245 11818769 13560180 18317490
   8265774 6309623 12040249 5067040 1629908 1856160 2143945
   13419460
3 1 7
4 7 15
5 10 3
6 4 9
7 12 5
8 3 5
9 1 2
10 12 14
11 3 4
12 11 4
13 8 2
14 13 1
15 1 3
16 1 6
```

**【样例 2 输出】**

```
1 17157248
```

**【样例 3】**

见附加文件中的 tree/ex\_tree3.in 与 tree/ex\_tree3.ans。

**【样例 4】**

见附加文件中的 tree/ex\_tree4.in 与 tree/ex\_tree4.ans。

**【样例 5】**

见附加文件中的 tree/ex\_tree5.in 与 tree/ex\_tree5.ans。

**【数据范围】**

对于所有数据, 保证  $1 \leq n \leq 100$ ,  $0 \leq a_i < m \leq 10^9$ ,  $1 \leq u, v \leq n$ 。



测试点编号	$n \leq$	特殊性质
1 ~ 3	15	无
4	30	A
5	30	B
6	30	C
7, 8	30	无
9, 10	100	A
11	100	B
12, 13	100	C
14 ~ 20	100	无

特殊性质 A：保证图为其中一端点为 1 号点的一条链。

特殊性质 B：保证图为以 1 为根的菊花图。

特殊性质 C：保证  $0 \leq a_i \leq 1$ 。

## 钻石岛 (island)

### 【题目描述】

《钻石岛》是一款 java 手机游戏，有一个  $1 \times 1 \times 2$  的长方体放置在二维平面上，每次你可以选择上下左右四个方向之一翻转该长方体。游戏的目标是经过一系列的翻转，让这个长方体移动到指定位置，且底面的形状与初始时相同。如下图：



(a)

在真正的游戏中，这个二维平面实际上是一个地图，也就是说，有的格子是不能使用的，在游戏中还有各种道具会影响长方体的移动。我们考虑一个更加纯粹的问题：初始时有一个  $a \times b \times c$  的长方体放置在一个无限大的二维平面上，底面四个点的坐标分别为  $(0,0), (a,0), (0,b), (a,b)$ ，也就是说，初始时长方体与  $x$  轴平行的棱长为  $a$ ，与  $y$  轴平行的棱长为  $b$ ，长方体的高是  $c$ 。每次可以选择上下左右四个方向之一翻转该长方体。游戏的目标是经过一系列的翻转，让长方体移动到指定位置  $(x,y), (a+x,y), (x,b+y), (a+x,b+y)$ ，也就是说，要求移动前后长方体的底面形状保持不变。你要求出移动到指定位置所需的最少步数。当然，并非对于所有的情况，都能把长方体移动到指定

位置，此时你应该报告无解。

**【输入格式】**

从文件 *island.in* 中读入数据。

输入一行，包含五个数， $a, b, c, x, y$ ，意义如上。

**【输出格式】**

输出到文件 *island.out* 中。

如果能移动到指定位置，输出一行一个整数，表示最少步数。否则输出 ‘impossible’。

**【样例 1 输入】**

1 1 1 2 9 0

**【样例 1 输出】**

1 6

**【样例 1 解释】**

就是《钻石岛》的第一关，也就是上面那张图。

**【样例 2 输入】**

1 3 4 5 8 0

**【样例 2 输出】**

1 2

**【样例 3 输入】**

1 3 4 5 -8 9

**【样例 3 输出】**

1 4

**【样例 4 输入】**

1 3 4 5 8 0

**【样例 4 输出】**

1 3 4 5 123 45

**【样例 5 输入】**

1 20 10 30 13 37

**【样例 5 输出】**

1 impossible

**【样例 6 输入】**

1 951 951 165 -373721873115219 -157139719882902

**【样例 6 输出】**

1 558214083389

**【数据范围】**

对于所有数据，保证  $1 \leq a, b, c \leq 1000, 0 \leq |x|, |y| \leq 10^{18}$ 。

子任务 1 (10 分):  $a, b, c, |x|, |y| \leq 10$ 。

子任务 2 (15 分):  $a, b, c, |x|, |y| \leq 100$ 。

子任务 3 (10 分): 保证  $a = b = c$ 。

子任务 4 (25 分): 保证  $a = b$ 。

子任务 5 (20 分):  $|x|, |y| \leq 10^5$ 。

子任务 6 (20 分): 无特殊限制。