

## 排序

Aizhan有一个由 $N$ 个互不相同的整数组成的序列 $S[0], S[1], \dots, S[N-1]$ , 其中 $S[i]$ 取值范围是 $[0, N-1]$ 。Aizhan试图通过交换某些元素对的方法将这个序列按照升序排序。Aizhan的朋友ErmeK也想交换某些元素对, ErmeK的交换未必有助于Aizhan的排序。

ErmeK和Aizhan打算通过若干轮次来修改这个序列。在每一轮, ErmeK首先做一次交换, 然后Aizhan做另一次交换。更确切地说, 做交换的人选择两个有效的下标并交换这两个下标的元素。请注意这两个下标可能相同。如果它们相等, 则对这个元素自身做交换, 并不改变这个序列。

Aizhan知道ErmeK并不关心对序列 $S$ 排序的事情。Aizhan还知道ErmeK将会选择哪些下标。ErmeK打算参加 $M$ 轮交换, 将这些轮次从0到 $M-1$ 编号。对于0到 $M-1$ 之间的每个 $i$ , ErmeK在第 $i$ 轮将选择下标 $X[i]$ 和 $Y[i]$ 的元素进行交换。

Aizhan要对序列 $S$ 按升序进行排序。在每一轮之前, 如果Aizhan看到当前的序列已经按升序排列, 她将结束这个排序过程。给定初始序列 $S$ 以及ErmeK要选择的下标, 请你找出一个交换的序列, 使得Aizhan能完成对序列 $S$ 的排序。此外, 在有些子任务中, 你还要找出尽可能短的交换序列来完成排序任务。题目保证通过 $M$ 或更少的轮次能够将序列 $S$ 排好序。

请注意如果Aizhan发现在ErmeK的交换之后, 序列 $S$ 已经排好序, 则Aizhan可以选择交换两个相同下标 (例如0和0) 的元素。这样, 序列 $S$ 在这一轮次之后也完成排序, 于是也达到了Aizhan的目标。另外, 如果初始序列 $S$ 就已经排好序, 那么所需的最少排序轮数就是0。

### 样例 1

设:

- 初始序列为 $S = 4, 3, 2, 1, 0$ 。
- ErmeK打算做 $M = 6$ 轮交换。
- ErmeK打算选择的下标序列 $X$ 和 $Y$ 分别是 $X = 0, 1, 2, 3, 0, 1$ 和 $Y = 1, 2, 3, 4, 1, 2$ 。换句话说, ErmeK打算交换的下标对是 $(0, 1), (1, 2), (2, 3), (3, 4), (0, 1)$ 和 $(1, 2)$ 。

按照上述设定, Aizhan能够通过三轮排序, 将序列 $S$ 排序为 $0, 1, 2, 3, 4$ 。这三轮排序所选择的下标分别是 $(0, 4), (1, 3)$ 和 $(3, 4)$ 。

下表给出了ErmeK和Aizhan修改这个序列的过程。

轮次	操作者	交换的下标对	序列
初态			4, 3, 2, 1, 0
0	Ermek	(0, 1)	3, 4, 2, 1, 0
0	Aizhan	(0, 4)	0, 4, 2, 1, 3
1	Ermek	(1, 2)	0, 2, 4, 1, 3
1	Aizhan	(1, 3)	0, 1, 4, 2, 3
2	Ermek	(2, 3)	0, 1, 2, 4, 3
2	Aizhan	(3, 4)	0, 1, 2, 3, 4

## 样例 2

设:

- 初始序列为  $S = 3, 0, 4, 2, 1$ 。
- Ermek打算做  $M = 5$  轮交换。
- Ermek打算选择的下标对是  $(1, 1)$ ,  $(4, 0)$ ,  $(2, 3)$ ,  $(1, 4)$  和  $(0, 4)$ 。  
按照上述设定, Aizhan能够通过三轮完成对序列  $S$  的排序。例如可通过选择下标对  $(1, 4)$ ,  $(4, 2)$  和  $(2, 2)$  来实现。下表给出了Ermek和Aizhan修改这一序列的过程。

轮次	操作者	交换的下标对	序列
初态			3, 0, 4, 2, 1
0	Ermek	(1, 1)	3, 0, 4, 2, 1
0	Aizhan	(1, 4)	3, 1, 4, 2, 0
1	Ermek	(4, 0)	0, 1, 4, 2, 3
1	Aizhan	(4, 2)	0, 1, 3, 2, 4
2	Ermek	(2, 3)	0, 1, 2, 3, 4
2	Aizhan	(2, 2)	0, 1, 2, 3, 4

## 任务

给定序列  $S$ 、 $M$  和下标序列  $X$  和  $Y$ , 请找出Aizhan对序列  $S$  完成排序所需的交换的序列。在子任务5—8中, 你找出的交换序列必须是最短的。

你需要实现函数 `findSwapPairs`:

- `findSwapPairs(N, S, M, X, Y, P, Q)` — grader调用这个函数刚好一次。
  - $N$ : 序列  $S$  的长度。
  - $S$ : 一个整数数组, 表示初始序列  $S$ 。
  - $M$ : Ermek打算做交换的次数。

- $X, Y$ : 长度为  $M$  的整数数组. 对于  $0 \leq i \leq M - 1$ , 在第  $i$  轮 Ermek 打算交换下标为  $X[i]$  和  $Y[i]$  的数组。
- $P, Q$ : 整数数组。利用这两个数组报告 Aizhan 完成对序列  $S$  排序的一种可能的交换序列, 假设这个交换序列的长度为  $R$ , 对于  $0$  到  $R - 1$  之间的每个  $i$ , Aizhan 在轮次  $i$  选择的下标将被存入  $P[i]$  和  $Q[i]$ 。你可以假设数组  $P$  和  $Q$  均已分别被分配了  $M$  个元素。
  - 这个函数应返回  $R$  的值(定义如上)。

## 子任务

子任务	得分	$N$	$M$	$X, Y$ 的额外限制	对 $R$ 的要求
1	8	$1 \leq N \leq 5$	$M = N^2$	$X[i] = Y[i] = 0$	$R \leq M$
2	12	$1 \leq N \leq 100$	$M = 30N$	$X[i] = Y[i] = 0$	$R \leq M$
3	16	$1 \leq N \leq 100$	$M = 30N$	$X[i] = 0, Y[i] = 1$	$R \leq M$
4	18	$1 \leq N \leq 500$	$M = 30N$	无	$R \leq M$
5	20	$1 \leq N \leq 2000$	$M = 3N$	无	最小
6	26	$1 \leq N \leq 200,000$	$M = 3N$	无	最小

数据保证存在一个仅需  $M$  或更少轮次的交换序列来完成排序。

### Sample grader

sample grader 从文件 `sorting.in` 中按照下列格式读入数据:

- 第 1 行:  $N$
- 第 2 行:  $S[0] \dots S[N - 1]$
- 第 3 行:  $M$
- 第 4 行, ...,  $M + 3$  行:  $X[i] \quad Y[i]$

sample grader 按下列格式输出:

- 第 1 行: findSwapPairs 函数的返回值  $R$
- 第  $2 + i$  ( $0 \leq i < R$ ) 行:  $P[i] \quad Q[i]$