

# 排序

Aizhan 有一個  $N$  個整數的數列  $S[0], S[1], \dots, S[N-1]$ . 數列由  $0$  到  $N-1$  的數字組成且互不相同。他試著透過數字的兩兩調換，將數列由小到大排序。他的朋友 Ermek 也會把幾對數字做兩兩調換，只是那未必是有幫助的。

Ermek 和 Aizhan 會進行幾輪的數列變換。在每一輪當中，Ermek 先調換一次，然後 Aizhan 再調換一次。說得更明確點，每次進行互換的人選出兩個有效的索引編號，並將該編號指向的元素互換。注意，這兩個索引編號有可能相同，當兩個索引編號相同，此人將元素和本身互換，也就不會改變數列。

Aizhan 知道 Ermek 並不在乎數列  $S$  的排序。他也知道 Ermek 會選的索引編號是哪個。Ermek 預計參與  $M$  輪的互換，我們將這幾輪編號為  $0$  到  $M-1$ ，對  $0$  到  $M-1$  之間的任一個  $i$ ，Ermek 在第  $i$  輪將會選擇索引編號  $X[i]$  和  $Y[i]$ 。

Aizhan 要將數列  $S$  排序。每一輪開始之前，如果 Aizhan 看到數列已經是由小到大排好，他將會終止整個過程。給定原來的數列  $S$  和 Ermek 會選擇的索引編號，你的任務是找出一系列的互換，幫助 Aizhan 完成排序。還有，在某些 subtasks 中，你找到的一系列互換必須愈短愈好。你可以假設數列  $S$  可以在  $M$  或更少次互換後完成排序。

注意，如果 Aizhan 看到 Ermek 的互換後，數列  $S$  已經依序排好，他可以選擇兩個相同的索引編號做互換(例如  $0$  和  $0$ )，結果就是這輪結束後，數列  $S$  也是依序排好，Aizhan 達到目的。並且注意，如果一開始數列  $S$  就已經依序排好，排序所需的輪數最小值為  $0$ 。

## Example 1

假設：

- 起始序列是  $S = 4, 3, 2, 1, 0$ .
- Ermek 願意做  $M = 6$  次互換。
- 代表 Ermek 將選擇的索引編號序列  $X$  和  $Y$  是  $X = 0, 1, 2, 3, 0, 1$  和  $Y = 1, 2, 3, 4, 1, 2$ ，換句話說，Ermek 打算選擇的各對索引編號是  $(0, 1)$ ,  $(1, 2)$ ,  $(2, 3)$ ,  $(3, 4)$ ,  $(0, 1)$ , 和  $(1, 2)$ 。

在這情況下，Aizhan 可以用三輪把數列  $S$  排序成  $0, 1, 2, 3, 4$ . 他可以選擇索引編號  $(0, 4)$ ,  $(1, 3)$  然後  $(3, 4)$ 。

下表顯示 Ermek 和 Aizhan 如何更改數列。

第幾輪	參與者	互換的索引編號	數列
開始時			4, 3, 2, 1, 0
0	Ermek	(0, 1)	3, 4, 2, 1, 0
0	Aizhan	(0, 4)	0, 4, 2, 1, 3

第幾輪	參與者	互換的索引編號	數列
1	Ernek	(1, 2)	0, 2, 4, 1, 3
1	Aizhan	(1, 3)	0, 1, 4, 2, 3
2	Ernek	(2, 3)	0, 1, 2, 4, 3
2	Aizhan	(3, 4)	0, 1, 2, 3, 4

## Example 2

假設:

- 起始序列是  $S = 3, 0, 4, 2, 1$ .
- Ernek 願意做  $M = 5$  次互換。
- Ernek 打算選擇的各對索引編號是 (1, 1), (4, 0), (2, 3), (1, 4), 和 (0, 4).

在這情況下，Aizhan 可以用三輪把數列  $S$  排序成例如，他可以選擇索引編號 (1, 4), (4, 2) 然後 (2, 2)。下表顯示 Ernek 和 Aizhan 如何更改數列。

第幾輪	參與者	互換的索引編號	數列
開始時			3, 0, 4, 2, 1
0	Ernek	(1, 1)	3, 0, 4, 2, 1
0	Aizhan	(1, 4)	3, 1, 4, 2, 0
1	Ernek	(4, 0)	0, 1, 4, 2, 3
1	Aizhan	(4, 2)	0, 1, 3, 2, 4
2	Ernek	(2, 3)	0, 1, 2, 3, 4
2	Aizhan	(2, 2)	0, 1, 2, 3, 4

## Task

你會拿到數列  $S$ ，數目  $M$ ，和索引編號序列  $X$  和  $Y$ 。計算出一個 Aizhan 用來完成排序的互換序列。在 subtasks 5 和 6，你找到的互換序列必須愈短愈好。

你必須實作函數 `findSwapPairs`:

- `findSwapPairs(N, S, M, X, Y, P, Q)` — 這個函數會被 grader 評分程式呼叫一次。
  - $N$ : 數列  $S$  的長度。
  - $S$ : 整數陣列，代表起始數列  $S$ 。
  - $M$ : Ernek 打算做互換的次數。
  - $X, Y$ : 長度  $M$  的整數陣列。For  $0 \leq i \leq M - 1$ , 在第  $i$  輪，Ernek 打算互換在索引編號  $X$  和  $Y$  的數字。
  - $P, Q$ : 整數陣列。用這些陣列來回報一個 Aizhan 可能用來排序  $S$  的互換序列。 $R$  代表你的程式找到的互換序列長度，對於  $0$  到  $R - 1$  之間的任一個  $i$ ，Aizhan 在第  $i$  輪所選擇的索引編號，必須存放在  $P[i]$  和  $Q[i]$ 。你可以假設  $P$  和  $Q$  陣列已

經各配置了  $M$  個元素。

- 這個函數必須回傳  $R$  的值(定義如上)。

## Subtasks

subtask	points	$N$	$M$	X, Y 的額外限制	R 須符合
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	$1 \leq N \leq 500$	$M = 30N$	無	$R \leq M$
5	20	$1 \leq N \leq 2000$	$M = 3N$	無	愈小愈好
6	26	$1 \leq N \leq 200,000$	$M = 3N$	無	愈小愈好

你可以假設只需要  $M$  或更少輪的解法是存在的。

## Sample grader

Sample grader 以下列格式從 `sorting.in` 檔案中讀入資料:

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

Sample grader 印出下列輸出。

- line 1: `findSwapPairs` 的回傳值  $R$ 。
- line  $2+i$ , for  $0 \leq i < R$ :  $P[i] \ Q[i]$