

Sorting 排序

Aizhen 有一個長度為 N 的整數序列 $S[0], S[1], \dots, S[N-1]$ 。這個序列由 0 至 $N-1$ 中不重覆的數字組成。他正在嘗試把這個序列中的數字通過兩兩交換的方式由小到大進行排序。而他的朋友 Ermek 也準備把序列中的數字兩兩交換 — 儘管是不必要的，但也會有幫助。

Ermek 和 Aizhen 正準備輪流地對序列進行交換。在每一輪中，首先由 Ermek 進行交換，然後由 Aizhen 來進行再一次交換。更確切地，進行一次交換就是先選出二個有效的序號然後把這二個序號中的元素交換。請注意這二個序號並不是一定不相同，如果它們是相同的，那個人就會對相同的元素進行交換，當然這不會改變這個序列。

Aizhen 知道 Ermek 實際上並不在乎序列 S 的排序結果，而他也知道 Ermek 打算選擇的一組序號。Ermek 計劃進行 M 輪交換。我們把它編號為 0 至 $M-1$ 。對於任何 i 在 0 至 $M-1$ 之間(含)，Ermek 會在第 i 輪選擇序號 $X[i]$ 和 $Y[i]$ 進行交換。

Aizhen 想把序列 S 排好順序。在每輪開始之前，如果 Aizhen 看見序列已經由小到大排好順序，他將會結束整個流程。現在給出初始序列 S 和 Ermek 打算選擇交換的序號，你的任務就是要找出一個序號交換的步驟，使得 Aizhen 能對序列 S 排好順序。此外，在一些子任務中你需要找出盡可能最短的步驟。你可以假定一定能在 M 輪或更少輪的交換就能使 S 排好順序。

注意，如果 Aizhen 看見 Ermek 交換之後序列 S 已經排好順序，他能選擇二個相同的序號進行交換(例如 0 和 0)。當然結果是序列 S 在整輪結束後仍然排好順序的，亦即 Aizhen 達到了他的目標。同樣地，如果初始序列 S 已經是排好順序的，則最少的交換輪數是 0 。

範例 1

假設如下：

- 初始序列 $S = 4, 3, 2, 1, 0$ 。
- Ermek 將會交換的次數 $M = 6$ 。
- 序列 X 和 Y 分別表示 Ermek 將會選擇交換的序號，它們是 $X = 0, 1, 2, 3, 0, 1$ 和 $Y = 1, 2, 3, 4, 1, 2$ 。換一種說法，Ermek 計劃交換的序號是 $(0, 1)$, $(1, 2)$, $(2, 3)$, $(3, 4)$, $(0, 1)$, 和 $(1, 2)$ 。

在這樣的前提下，Aizhen 能用三輪就把序列 S 排好順序 $0, 1, 2, 3, 4$ 。他能選擇序號 $(0, 4)$, $(1, 3)$ 和 $(3, 4)$ 進行交換。

下列的表格將會表示ErmeK和Aizhen如何對序列的修改：

| 輪數 | 參與者 | 被交換的二個序號 | 序列 |
|----|--------|----------|---------------|
| 初始 | | | 4, 3, 2, 1, 0 |
| 0 | ErmeK | (0, 1) | 3, 4, 2, 1, 0 |
| 0 | Aizhen | (0, 4) | 0, 4, 2, 1, 3 |
| 1 | ErmeK | (1, 2) | 0, 2, 4, 1, 3 |
| 1 | Aizhen | (1, 3) | 0, 1, 4, 2, 3 |
| 2 | ErmeK | (2, 3) | 0, 1, 2, 4, 3 |
| 2 | Aizhen | (3, 4) | 0, 1, 2, 3, 4 |

範例 2

假設如下：

- 初始序列 $S = 3, 0, 4, 2, 1$ 。
- ErmeK 將會交換的次數 $M = 5$ 。
- ErmeK 計劃交換的序號對是 (0, 1), (1, 2), (2, 3), (3, 4), (0, 1), 和 (1, 2) (1, 1), (4, 0), (2, 3), (1, 4), 和 (0, 4)。

在這樣的前提下，Aizhen 能用三輪就把序列 S 排好順序 0, 1, 2, 3, 4。例如可以選擇交換 (1, 4), (4, 2), 和 (2, 2)。下列的表格將會表示 ErmeK 和 Aizhen 如何對序列的修改：

| 輪數 | 參與者 | 被交換的二個序號 | 序列 |
|----|--------|----------|---------------|
| 初始 | | | 3, 0, 4, 2, 1 |
| 0 | ErmeK | (1, 1) | 3, 0, 4, 2, 1 |
| 0 | Aizhen | (1, 4) | 3, 1, 4, 2, 0 |
| 1 | ErmeK | (4, 0) | 0, 1, 4, 2, 3 |
| 1 | Aizhen | (4, 2) | 0, 1, 3, 2, 4 |
| 2 | ErmeK | (2, 3) | 0, 1, 2, 3, 4 |
| 2 | Aizhen | (2, 2) | 0, 1, 2, 3, 4 |

任務

給出序列 S , 次數 M 和序號的序列 X and Y . 請計算出 Aizhen 能把序列 S 排好順序的交換步驟，在子任務 5 – 6 中交換的輪數必需是最少的。

你需要編寫函數 `findSwapPairs`：

- `findSwapPairs(N, S, M, X, Y, P, Q)` — 這個函數只會被grader調用一次。
 - N : 序列 S 的長度。
 - S : 初始序列 S 。
 - M : ErmeK 計劃進行交換的次數。

- X, Y : 長度為 M 的整數數組。對於 $0 \leq i \leq M - 1$, 在第 i 輪 Ermek 計劃交換的序號為 $X[i]$ 和 $Y[i]$ 。
- P, Q : 整數的數組。這二個數組用來表示 Aizhen 能對序列 S 排好順序的其中一個交換步驟。其中 R 表示你的程序中找出的交換步驟的輪數。對於任何 i 在 0 和 $R - 1$ 之間 (含) , Aizhen 在第 i 輪選擇交換的序號應該存放在

$P[i]$ and $Q[i]$ 內。你可以假定數組 P 和 Q 已經各分配了 M 個元素。

- 這個函數應該返回 R 的數目。(上述已經定義)

子任務

| 子任務 | 分數 | N | M | 附加條件 X, Y | 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 | $1 \leq N \leq 500$ | $M = 30N$ | 沒有 | $R \leq M$ |
| 5 | 20 | $1 \leq N \leq 2000$ | $M = 3N$ | 沒有 | 盡可能少 |
| 6 | 26 | $1 \leq N \leq 200,000$ | $M = 3N$ | 沒有 | 盡可能少 |

你可以假定必定存在一個只需要 M 輪或更少輪的答案。

Sample grader

sample grader 讀取的輸入檔案 `sorting.in` 將會是以下的格式:

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

Sample grader 會輸出 `findSwapPairs` 的返回數目。