



Ordenamiento

Aizhan tiene una secuencia de N enteros $S[0], S[1], \dots, S[N - 1]$. La secuencia consiste en números distintos entre 0 y $N - 1$. Ella está intentando ordenar esta secuencia en orden ascendente intercambiando de lugar algunos pares de elementos. Su amigo Ermek también intercambiará de lugar algunos pares de elementos — no necesariamente de una manera útil.

Ermek y Aizhan modificarán la secuencia en una serie de rondas. En cada ronda, Ermek hace primero un intercambio y luego Aizhan hace otro. Más precisamente; la persona que hace un intercambio elige dos índices válidos e intercambia los elementos en esos índices. Toma en cuenta que los dos índices no tienen que ser necesariamente diferentes. Si son iguales, la persona haciendo el intercambio, cambia un elemento con si mismo, lo que no modifica la secuencia.

Aizhan sabe que a Ermek en realidad no le interesa ordenar la secuencia S . Ella también sabe exactamente los índices que elegirá Ermek. Ermek planea participar en M rondas de intercambios. Numeramos estas rondas de 0 a $M - 1$. Para cada i entre 0 y $M - 1$ incluidos, Ermek elegirá los índices $X[i]$ y $Y[i]$ en la ronda i .

Aizhan quiere ordenar la secuencia S . Antes de cada ronda, si Aizhan ve que la secuencia ya está en orden ascendente, terminará todo el proceso. Dadas la secuencia original S y los índices que Ermek escogerá, tu tarea es encontrar una secuencia de intercambios, los cuales puedan ser usados por Aizhan para ordenar la secuencia S . Además, en algunas subtarefas se te pedirá que encuentres una secuencia de intercambios que sea lo más corta posible. Puedes asumir que es posible ordenar la secuencia S en M o menos rondas.

Toma en cuenta que si Aizhan ve que la secuencia S está ordenada después del intercambio de Ermek, ella puede elegir intercambiar dos índices iguales (por ejemplo, 0 y 0). Como resultado la secuencia S también está ordenada después de que se completa la ronda, por lo que Aizhan logra su cometido. También hay que tomar en cuenta que si la secuencia S inicial ya está ordenada, el mínimo número necesario de rondas para ordenarla es 0 .

Ejemplo 1

Suponga que:

- La secuencia inicial es $S = 4, 3, 2, 1, 0$.
- Ermek quiere hacer $M = 6$ intercambios.
- Las secuencias X y Y que indican los índices que Ermek escogerá son $X = 0, 1, 2, 3, 0, 1$ y $Y = 1, 2, 3, 4, 1, 2$. En otras palabras, los pares de índices que Ermek planea escoger son $(0, 1)$, $(1, 2)$, $(2, 3)$, $(3, 4)$, $(0, 1)$, y $(1, 2)$.

En esta acomodación Aizhan puede ordenar la secuencia S en el orden $0, 1, 2, 3, 4$ en tres rondas. Ella puede hacerlo eligiendo los índices $(0, 4)$, $(1, 3)$, y luego $(3, 4)$.

La tabla siguiente muestra cómo Ermek y Aizhan modifican la secuencia.

Ronda	Jugador	Par de índices intercambiados	Secuencia
inicio			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

Ejemplo 2

Suponga que:

- La secuencia inicial es $S = 3, 0, 4, 2, 1$.
- Ermek quiere hacer $M = 5$ intercambios.
- Los pares de índices que Ermek planea escoger son $(1, 1)$, $(4, 0)$, $(2, 3)$, $(1, 4)$, y $(0, 4)$.

En esta acomodación Aizhan puede ordenar la secuencia S en tres rondas, por ejemplo escogiendo los pares de índices $(1, 4)$, $(4, 2)$, y luego $(2, 2)$. La tabla siguiente muestra cómo Ermek y Aizhan modifican la secuencia.

Ronda	Jugador	Par de índices intercambiados	Secuencia
inicio			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

Tarea

Se te dan la secuencia S , el número M , y las secuencias de índices X y Y . Encuentra una secuencia de intercambios, la cual Aizhan pueda usar para ordenar la secuencia S . En los subtasks 5 y 6 la secuencia de intercambios que encuentres debe ser lo más corta posible.

Necesitas implementar la función `findSwapPairs`:

- `findSwapPairs(N, S, M, X, Y, P, Q)` — Esta función será llamada por el calificador (grader) exactamente una vez.
 - N : la longitud de la secuencia S .

- S : un arreglo de enteros que contiene la secuencia inicial S .
- M : el número de intercambios que Ermek planea hacer.
- X, Y : arreglos de enteros de longitud M . Para $0 \leq i \leq M - 1$, en la ronda i Ermek planea intercambiar los números en los índices $X[i]$ y $Y[i]$.
- P, Q : arreglos de enteros. Usa este arreglo para reportar una posible secuencia de intercambios que Aizhan puede hacer para ordenar la secuencia S . Indique con R el largo de la secuencia de intercambios que tu programa ha encontrado. Para cada i entre 0 y $R - 1$ incluidos, los índices que Aizhan debería escoger en la ronda i deberían estar almacenados en $P[i]$ y $Q[i]$. Debes asumir que los arreglos P y Q ya tienen tamaño suficiente para almacenar hasta M elementos cada uno.
- Esta función debería retornar el valor de R (especificado antes).

Subtareas

subtarea	puntos	N	M	restricciones adicionales en X, Y	requerimiento en R
1	8	$1 \leq N \leq 5$	$M = N^2$	$X[i] = Y[i] = 0$ para todo i	$R \leq M$
2	12	$1 \leq N \leq 100$	$M = 30N$	$X[i] = Y[i] = 0$ para todo i	$R \leq M$
3	16	$1 \leq N \leq 100$	$M = 30N$	$X[i] = 0, Y[i] = 1$ para todo i	$R \leq M$
4	18	$1 \leq N \leq 500$	$M = 30N$	ninguna	$R \leq M$
5	20	$6 \leq N \leq 2000$	$M = 3N$	ninguna	mínimo posible
6	26	$6 \leq N \leq 200,000$	$M = 3N$	ninguna	mínimo posible

Puedes asumir que existe una solución que requiere M o menos rondas.

Calificador de ejemplo

El calificador de ejemplo lee la entrada del archivo `sorting.in` en el siguiente formato:

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

El calificador de ejemplo imprime la siguiente salida:

- línea 1: el valor R que retorna `findSwapPairs`
- línea $2+i$, para $0 \leq i < R$: $P[i] Q[i]$