



Sorting

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

Ermek y Aizhan están modificando la secuencia en una serie de rondas. En cada ronda, primero Ermek hace un intercambio y entonces Aizhan hace otro intercambio. Más precisamente, la persona que hace un intercambio elige dos índices válidos e intercambia los elementos en esos índices. Note que los dos índices no tienen que ser distintos. Si ellos son iguales, la persona actual intercambia un elemento con el mismo, lo cual no cambia la secuencia.

Aizhan sabe que Ermek realmente no tiene cuidado ordenando la secuencia S . Ella también sabe los índices exactos que Ermek elige. Ermek tiene pensado tomar parte en M rondas de intercambios. Nosotros numeraremos estas rondas desde 0 a $M - 1$. Para cada i entre 0 y $M - 1$ inclusive, Ermek eligirá los índices de $X[i]$ 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 ordenada en orden ascendente, ella terminará el proceso entero. Dada la secuencia original S y los índices que Ermek eligió, tu tarea es encontrar una secuencia de intercambios, que Aizhan puede usar para ordenar la secuencia S . En adición, en algunas subtarefas se requiere encontrar una secuencia de intercambios que sea tan corta como sea posible. Usted puede asumir que es posible ordenar la secuencia S en M o menos rondas.

Note que 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 un resultado la secuencia S está también ordenada después de una ronda entera, así que Aizhan alcanza su objetivo. También note que si la secuencia inicial S ya está ordenada, el número mínimo de rondas necesarias para ordenar esta es 0 .

Ejemplo 1

Suponga que:

- La secuencia inicial es $S = 4, 3, 2, 1, 0$.
- Ermek está deseando hacer $M = 6$ intercambios.
- La secuencia X y Y que describe los índices de Ermek 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 elegir son $(0, 1)$, $(1, 2)$, $(2, 3)$, $(3, 4)$, $(0, 1)$, y $(1, 2)$.

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

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

Ronda	Jugador	Pares de indices intercambiados	Secuencia
comenzando			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 esta deseando hacer $M = 5$ intercambios.
- Los pares de indices que Ermek planea elegir son $(1, 1)$, $(4, 0)$, $(2, 3)$, $(1, 4)$, y $(0, 4)$.

En esta configuracio Aizhan puede ordenar la secuencia S en tres rondas, por ejemplo eligiendo los pares de indices $(1, 4)$, $(4, 2)$, y entonces $(2, 2)$. La siguiente tabla muestra como Ermek y Aizhan modifican la secuencia.

Ronda	Jugador	Pares de indices intercambiados	Secuencia
beginning			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

A usted se le da la secuencia S , el numero M , y la secuencia de indices X y Y . Encuentre una secuencia de intercambios, los cuales Aizhan puede usar para ordenar la secuencia S . En las subtareas 5 – 8 la secuencia de intercambios que usted puede encontrar tiene que ser la menor posible..

Usted necesita implementar la funcion `findSwapPairs`:

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

- S : la secuencia inicial S .
- M : el numero 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 que Ermek planea intercambiar los numeros en los indices $X[i]$ y $Y[i]$.
- P, Q : arreglos de enteros. Use estos arreglos para reportar una posible secuencia de intercambios que Aizhan puede hacer para ordenar la secuencia S . Denote por R la longitud de la secuencia de intercambios que su programa tiene que encontrar. Para cada i entre 0 y $R - 1$ inclusive, los indices que Aizhan debe elegir en la ronda i deben ser almacenados en $P[i]$ y $Q[i]$. Usted puede asumir que los arreglos P y Q son de tamaño M .
 - Esta funcion debe retornar los valores de R (definidos arriba).

Subtareas

subtarea	puntos	N	M	restricciones extra en X, Y	requerimientos de R
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	$1 \leq N \leq 500$	$M = 30N$	none	$R \leq M$
5	20	$1 \leq N \leq 2000$	$M = 3N$	none	minimo posible
6	26	$1 \leq N \leq 200,000$	$M = 3N$	none	minimo posible

Usted puede asumir que existe una solucion que requiere M o menos rounds.

Ejemplo de grader

El grader de ejemplo lee la entrada desde el fichero `sorting.in` en el siguiente formato:

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

El ejemplo de grader imprime el valor de `ret`The sample grader prints the return value of `findSwapPairs`.