



Tri

Aizhan dispose d'une séquence de N entiers $S[0], S[1], \dots, S[N - 1]$. La séquence est constituée de nombres distincts de 0 à $N - 1$. Elle essaie de trier cette séquence dans un ordre croissant en échangeant des paires d'éléments. Son amie Ermek effectuera aussi des échanges de paires d'éléments — pas nécessairement dans le but de l'aider.

Ermek et Aizhan vont modifier la séquence en une série de tours. À chaque tour, Ermek fait un échange en premier puis Aizhan effectue un autre échange. Plus précisément, la personne effectuant un échange choisit deux indices valides et échange les éléments de ces deux indices. Notez que les deux indices ne doivent pas être forcément distincts. S'ils sont égaux, la personne permute un élément avec lui-même, ce qui n'affecte en rien la séquence.

Aizhan sait que Ermek ne s'intéresse pas réellement au tri de la séquence S . Elle sait aussi exactement quels seront les indices choisis par Ermek. Ermek prévoit d'effectuer M échanges. On énumère ces échanges de 0 à $M - 1$. Pour chaque i entre 0 et $M - 1$ inclus, Ermek choisira les indices $X[i]$ et $Y[i]$ du tour i .

Aizhan désire trier la séquence S . Avant chaque tour, si Aizhan voit que la séquence est déjà triée dans un ordre croissant, elle terminera la séquence d'actions. Etant donné la séquence initiale S et les indices qu'Ermek choisira, votre tâche est de trouver une séquence d'échanges qu'Aizhan pourra effectuer pour trier la séquence S . En plus, dans certaines sous-tâches vous devez trouver une plus courte séquence d'échanges possible. Vous pouvez supposer qu'il est possible de trier la séquence S en M tours ou moins.

Notez que si Aizhan voit que la séquence S est triée juste après l'échange effectué par Ermek, elle pourra choisir d'échanger deux indices identiques (par ex. : 0 et 0). Ainsi, la séquence S sera toujours triée à la fin du tour, et Aizhan aura atteint son but. De plus, notez que si la séquence S est initialement triée, le nombre minimal de tours nécessaires est égal à 0 .

Exemple 1

Supposons que :

- La séquence initiale est $S = 4, 3, 2, 1, 0$.
- Ermek veut effectuer $M = 6$ échanges.
- Les séquences X et Y qui décrivent les indices qu'Ermek choisira sont $X = 0, 1, 2, 3, 0, 1$ et $Y = 1, 2, 3, 4, 1, 2$. En d'autres termes, les paires d'indices qu'Ermek veut choisir sont $(0, 1)$, $(1, 2)$, $(2, 3)$, $(3, 4)$, $(0, 1)$, et $(1, 2)$.

Avec cette configuration, Aizhan peut trier la séquence S dans l'ordre $0, 1, 2, 3, 4$ en trois tours. Elle pourra faire ceci en choisissant les indices $(0, 4)$, $(1, 3)$ et $(3, 4)$.

La table suivante montre comment Ermek et Aizhan modifient la séquence.

Tour	Joueur	Paire d'indices échangés	Séquence
Départ			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

Exemple 2

Supposons que :

- La séquence initiale est $S = 3, 0, 4, 2, 1$.
- Ermek veut effectuer $M = 5$ échanges.
- Les séquences X et Y qui décrivent les indices qu'Ermek veut choisir sont $(1, 1)$, $(4, 0)$, $(2, 3)$, $(1, 4)$ et $(0, 4)$.

Avec cette configuration, Aizhan peut trier la séquence S en trois tours, par exemple en choisissant les paires d'indices $(1, 4)$, $(4, 2)$ et $(2, 2)$. La table suivante montre comment Ermek et Aizhan modifient la séquence.

Tour	Joueur	Paire d'indices échangés	Séquence
Départ			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

Tâche

On vous fournit la séquence S , le nombre M et les séquences d'indices X et Y . Retrouvez une séquence d'échanges qu'Aizhan pourra utiliser pour trier la séquence S . Au niveau des sous-tâches 5 et 6, la séquence d'échanges que vous trouvez doit être la plus courte possible.

Vous devez implémenter la fonction `findSwapPairs` :

- `findSwapPairs(N, S, M, X, Y, P, Q)` — appelée par l'évaluateur exactement une fois
 - N : la longueur de la séquence S .
 - S : un tableau d'entiers contenant la séquence initiale S .

- M : le nombre d'échanges qu'ErmeK prévoit d'effectuer.
- X, Y : tableaux d'entiers de longueur M . Pour $0 \leq i \leq M - 1$, au tour i ErmeK prévoit d'échanger les nombres aux indices $X[i]$ et $Y[i]$.
- P, Q : tableaux d'entiers. Utilisez ces tableaux pour fournir une séquence possible d'échanges à effectuer pour trier la séquence S . Considérons que R est la longueur de la séquence de permutations que votre programme vient de trouver. Pour chaque i entre 0 et $R - 1$ inclus, les indices qu'Aizhan devra choisir lors du tour i doivent être stockés dans $P[i]$ et $Q[i]$. Vous devez supposer que les tableaux P et Q sont déjà alloués à une taille de M éléments chacun.
- Cette fonction doit renvoyer la valeur de R (définie précédemment).

Sous-tâches

sous-tâche	points	N	M	Contraintes supplémentaires sur X, Y	Condition sur R
1	8	$1 \leq N \leq 5$	$M = N^2$	$X[i] = Y[i] = 0$ pour tout i	$R \leq M$
2	12	$1 \leq N \leq 100$	$M = 30N$	$X[i] = Y[i] = 0$ pour tout i	$R \leq M$
3	16	$1 \leq N \leq 100$	$M = 30N$	$X[i] = 0, Y[i] = 1$ pour tout i	$R \leq M$
4	18	$1 \leq N \leq 500$	$M = 30N$	Aucune	$R \leq M$
5	20	$6 \leq N \leq 2000$	$M = 3N$	Aucune	minimum possible
6	26	$6 \leq N \leq 200000$	$M = 3N$	Aucune	minimum possible

Vous pouvez supposer qu'il existe une solution qui nécessite M tours ou moins.

L'évaluateur fourni (Grader)

L'évaluateur d'exemple lit l'entrée depuis le fichier `sorting.in` suivant ce format:

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

L'évaluateur d'exemple écrit dans le format suivant :

- ligne 1 : la valeur de retour R de `findSwapPairs`.
- ligne $2+i$, pour $0 \leq i < R$: $P[i] \ Q[i]$