

Rendezés

Aizhannak van egy N elemű $S[0], S[1], \dots, S[N - 1]$ számsorozata, amely elemei mind különböző 0 és $N - 1$ közötti számok. Növekvő sorrendbe szeretné rendezni a sorozatot, elemek cseréjével. A barátja, Ermek is cserélget néhány elempárt, de ezzel nem biztos hogy segít neki.

M menetet hajtanak végre, minden menetben először Ermek, utána Aizhan cserél. Aizhan előre tudja, hogy Ermek mely elempárokat fogja megcserélni, az i -edik menetben az $X[i]$ -t és az $Y[i]$ -t, $i = 0, \dots, M - 1$ -re. Ha Aizhan egy forduló előtt azt látja, hogy már rendezett a sorozat, akkor be kell fejeznie a folyamatot!

A feladatod megkeresni Aizhan cseresorozatát. Néhány részfeladatnál ennek a cserék számának minimálisnak kell lenni! Feltehető, hogy az S sorozat rendezhető M vagy kevesebb lépésben.

Megjegyzendő, hogy ha Aizhan azt látja, hogy a sorozat rendezett Ermek cseréje után, akkor azonos elemeket cserélhet (pl. a 0 -t a 0 -val). Ha a sorozat már kezdetben is rendezett, akkor a menetek minimális száma 0 .

1. példa

Legyen:

- a kezdő sorozat $S = 4, 3, 2, 1, 0$.
- Ermek cseréi maximális száma $M = 6$.
- Az X és Y sorozat írja le Ermek cseréit, $X = 0, 1, 2, 3, 0, 1$ és $Y = 1, 2, 3, 4, 1, 2$. Azaz Ermek cseréi a $(0, 1)$, $(1, 2)$, $(2, 3)$, $(3, 4)$, $(0, 1)$ és $(1, 2)$ indexű elempárok.

Ebben az esetben Aizhan az S sorozatot a kívánt $0, 1, 2, 3, 4$ sorrendbe 3 menetben rendezheti. Aizhan cseréi: $(0, 4)$, $(1, 3)$ és $(3, 4)$ indexű elempárok.

A táblázat mutatja a cserék hatását.

Menet	Játékos	csereindexek	sorozat
kezdetben			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

2. példa

Legyen:

- a kezdősorozat $S = 3, 0, 4, 2, 1$.
- Ernek $M = 5$ cserét akar végrehajtani.
- Ernek tervezett cseréi $(1, 1)$, $(4, 0)$, $(2, 3)$, $(1, 4)$ és $(0, 4)$.

Ekkor Aizhan három menetben rendezheti a sorozatot, pl. $(1, 4)$, $(4, 2)$ és $(2, 2)$. A táblázat mutatja a cserék hatását.

Menet	Játékos	csereindexek	sorozat
kezdetben			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

Feladat

Adott S, M, X és Y esetén számítsd ki, hogy Aizhan milyen elempárok cseréjével éri el a rendezést! Az 5 – 6. részfeladat esetén a cserék számának minimálisnak kell lenni!

A `findSwapPairs` függvényt kell megvalósítanod:

- `findSwapPairs(N, S, M, X, Y, P, Q)` — Az értékelő pontosan egyszer hívja.
 - N : a rendezendő S sorozat hossza.
 - S : a rendezendő S sorozat.
 - M : Ernek tervezett cseréi száma.
 - X, Y : M elemű tömbök. Minden $0 \leq i \leq M - 1$ -re, az i . menetben Ernek tervezett cseréje $X[i]$ és $Y[i]$.
 - P, Q : egész számok tömbjei, amellyel Aizhan megoldását közöld. Jelölje R Aizhan megoldásához szükséges menetek számát, minden i -re 0 és $R - 1$ között az i . menetben Aizhan a $P[i]$ és $Q[i]$ indexű elemeket cseréli. Feltehető, hogy a P és Q tömbök M eleműek.
 - A függvény visszaadott értéke R legyen (lásd fent)!

Részfeladatok

részfeladat	pont	N	M	korlátok X, Y -ra	feltételek R -re
1	8	$1 \leq N \leq 5$	$M = N^2$	$X[i] = Y[i] = 0$	$R \leq M$

részfeladat	pont	$1 \leq N \leq 100$	$M = 30N$	korlátok X, Y-ra $X[i] = Y[i] = 0$	feltételek R-re $R \leq M$
3	16	$1 \leq N \leq 100$	$M = 30N$	$X[i] = 0, Y[i] = 1$	$R \leq M$
4	18	$1 \leq N \leq 500$	$M = 30N$	nincs	$R \leq M$
5	20	$1 \leq N \leq 2000$	$M = 3N$	nincs	minimális
6	26	$1 \leq N \leq 200,000$	$M = 3N$	nincs	minimális

Feltehető, hogy az S sorozat rendezhető M vagy kevesebb lépésben.

Megvalósítás

A `sorting.c`, `sorting.cpp`, `sorting.pas`, or `sorting.java` file-t kell beküldened, ami a megoldásod tartalmazza.

C/C++ program (include `sorting.h` legyen a file elején)

```
int findSwapPairs(int N, int S[], int M, int X[], int Y[], int P[], int C
```

Pascal programs (implement the described method in the unit `sorting`)

```
function findSwapPairs(N: longint; var S: array of longint, var M: longir
```

Java programs (implement the described method in the public class

`sorting`)

```
public int findSwapPairs(int N, int S[], int M, int X[], int Y[], int P[]
```

Minta értékelő

A minta értékelő a `sorting.in` file-t olvassa:

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

A mintaértékelő kiírja a `findSwapPairs` értékét.