

Sortowanie (Sorting)

Aizhan pragnie posortować ciąg N liczb całkowitych $S[0], S[1], \dots, S[N - 1]$. Ciąg ten składa się z parami różnych liczb z zakresu od 0 do $N - 1$. Aizhen chce posortować zadany ciąg rosnąco zamieniając elementy ciągu parami. Jej przyjaciel Ermek ma także zamiar zamieniać parami niektóre elementy ciągu, ale niekoniecznie w sposób, który pomagałby Aizhan.

Ermek i Aizhan modyfikują ciąg w rundach. W każdej rundzie pierwszą zamianę wykonuje Ermek, a po nim zamiany dokonuje Aizhan. Dokładniej, osoba dokonująca zamiany wybiera dwa indeksy, po czym zamienia miejscami elementy o tych indeksach. Indeksy wybrane do zamiany niekoniecznie muszą być różne — jeśli oba są takie same, oznacza to, że w rzeczywistości element pozostaje na swoim miejscu.

Aizhan wie, że Ermekowi nie zależy na posortowaniu ciągu. Ponadto wie ona, jakich zamian Ermek ma zamiar dokonać. Ermek planuje wziąć udział w co najwyżej M rundach zamian elementów ciągu. Rundy są ponumerowane od 0 do $M - 1$. W i -tej rundzie, dla $i = 0, 1, \dots, M - 1$, Ermek wybiera do zamiany elementy o indeksach $X[i]$ oraz $Y[i]$.

Aizhan chce posortować ciąg S . Przed każdą rundą, jeśli Aizhan widzi, że ciąg jest już posortowany rosnąco, to przerywa cały proces sortowania. Mając dany ciąg S oraz listę zamian, które chce wykonać Ermek, znajdź ciąg zamian dla Aizhan, który pozwoli jej posortować ciąg S . Dodatkowo, w niektórych podzadaniach musisz wyznaczyć najkrótszy taki ciąg. Możesz założyć, że zawsze możliwe jest posortowanie ciągu S w M lub mniej rundach.

Zauważ, że jeśli Aizhan widzi, że ciąg S jest posortowany po zamianie Ermeka, może do zamiany wskazać elementy o tym samym indeksie (np. 0 i 0). W wyniku takiej zamiany ciąg S pozostaje posortowany po całej rundzie, tak więc Aizhan osiąga swój cel. Zauważ także, że jeżeli początkowy ciąg S jest posortowany, to minimalna liczba rund potrzebnych do posortowania go wynosi 0 .

Przykład 1

Założmy, że:

- Początkowy ciąg to $S = 4, 3, 2, 1, 0$.
- Ermek chce wykonać $M = 6$ zamian.
- Ciągi X i Y opisujące pary indeksów elementów zamienianych przez Ermeka to $X = 0, 1, 2, 3, 0, 1$ oraz $Y = 1, 2, 3, 4, 1, 2$. Innymi słowy, pary indeksów elementów, które Ermek zamierza zamienić ze sobą to kolejno: $(0, 1)$, $(1, 2)$, $(2, 3)$, $(3, 4)$, $(0, 1)$, i $(1, 2)$.

W tej sytuacji Aizhan może posortować S , otrzymując w trzech rundach kolejność $0, 1, 2, 3, 4$. Osiągnie to wybierając pary indeksów $(0, 4)$, $(1, 3)$, a potem $(3, 4)$.

Poniższa tabela pokazuje, jak będzie wyglądał ciąg po zamianach Ermeka i Aizhan:

Runda	Gracz	Pary indeksów	Ciąg
początek			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

Przykład 2

Założmy, że:

- Początkowy ciąg to $S = 3, 0, 4, 2, 1$.
- Ermek chce wykonać $M = 5$ zamian.
- Pary indeksów elementów, które Ermek zamierza zamienić ze sobą to kolejno: $(1, 1)$, $(4, 0)$, $(2, 3)$, $(1, 4)$, i $(0, 4)$.

W tej sytuacji Aizhan może posortować S w trzech rundach, wybierając na przykład pary indeksów $(1, 4)$, $(4, 2)$, oraz $(2, 2)$. Poniższa tabela pokazuje, jak będzie wyglądał ciąg po zamianach Ermeka i Aizhan:

Runda	Gracz	Pary indeksów	Ciąg
początek			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

Zadanie

Dane są: ciąg S , liczba M , oraz listy indeksów X i Y . Znajdź kolejność zamian, których Aizhan może użyć do posortowania ciągu S . W podzadaniach 5 i 6 Twój ciąg zamian dodatkowo musi być najkrótszy z możliwych.

Musisz zaimplementować funkcję `findSwapPairs`:

- `findSwapPairs(N, S, M, X, Y, P, Q)` — Funkcja ta zostanie wykonana przez program sprawdzający dokładnie raz.
 - N : długość ciągu S .

- S : tablica zawierająca początkowy ciąg S .
- M : liczba zamian, które chce wykonać Ermek.
- X, Y : tablice liczb całkowitych długości M . Dla $0 \leq i \leq M - 1$, Ermek w rundzie i zamieni elementy ciągu S o indeksach $X[i]$ oraz $Y[i]$.
- P, Q : tablice liczb całkowitych. Użyj tych tablic, aby podać jeden z możliwych ciągów zamian, których może użyć Aizhan do posortowania ciągu S . Niech R będzie długością ciągu zamian znalezionej przez Twój program. Dla każdego i między 0 a $R - 1$ włącznie, indeksy zamienione przez Aizhan powinny zostać zapisane w $P[i]$ oraz $Q[i]$. Możesz założyć, że tablice P i Q zostały już zaalokowane i każda z nich ma długość M .
- Funkcja ta powinna zwrócić wartość R (zdefiniowaną powyżej).

Podzadania

podzadanie	punkty	N	M	dotatkowe ograniczenia na X, Y	wymagane R
1	8	$1 \leq N \leq 5$	$M = N^2$	$X[i] = Y[i] = 0$ dla wszystkich i	$R \leq M$
2	12	$1 \leq N \leq 100$	$M = 30N$	$X[i] = Y[i] = 0$ dla wszystkich i	$R \leq M$
3	16	$1 \leq N \leq 100$	$M = 30N$	$X[i] = 0, Y[i] = 1$ dla wszystkich i	$R \leq M$
4	18	$1 \leq N \leq 500$	$M = 30N$	brak	$R \leq M$
5	20	$1 \leq N \leq 2000$	$M = 3N$	brak	najmniejsze możliwe
6	26	$1 \leq N \leq 200,000$	$M = 3N$	brak	najmniejsze możliwe

Możesz założyć, że istnieje rozwiązanie, które wymaga M lub mniej rund.

Przykładowy program sprawdzający

Przykładowy program sprawdzający czyta dane z pliku `sorting.in` w następującej postaci:

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

Program wypisuje na wyjściu kolejno:

- wiersz 1: wartość zwróconą przez funkcję `findSwapPairs`
- wiersz $2+i$ dla $0 \leq i \leq R$: $P[i] \ Q[i]$