

## Sorteerimine

Aižanil on  $N$ -elemendiline jada  $S[0], S[1], \dots, S[N - 1]$ , mis koosneb kõigist erinevatest numbritest  $0$ -st kuni  $N - 1$ -ni. Aižan üritab sorteerida selle jada kasvavasse järjekorda, vahetades numbreid omavahel paarikaupa. Tema sõber Ermek osaleb selles protsessis, teostades ka mõned paaride vahetused, aga mitte tingimata Aižanile kasulikul viisil.

Ermek ja Aižan hakkavad jada muutma voorude kaupa. Igas voorus teostab paari vahetuse esimesena Ermek, järgmisena teeb oma vahetuse Aižan. Vahetuse teostaja valib jadas kaks indeksit ning vahetab nende indeksite kohal asuvad elemendid. Valitud indeksid ei pea olema erinevad. Kui nad on võrdsed, toimub vastava elemendi “vahetus” iseendaga ehk siis jada ei muutu. Niipea kui mõne vooru alguses on jada sorteeritud, protsess lõpeb.

Aižan teab, et Ermek ei hooli tegelikult jada  $S$  järjestamisest. Õnneks aga teab ta iga vooru jaoks kõik täpsed indeksid ette, mida Ermek plaanib valida. Täpsemalt, Ermek plaanib võtta osa maksimaalselt  $M$  voorust. Nummerdame neid  $0$  kuni  $M - 1$ . Iga  $0 \leq i < M$  puhul valib Ermek oma käigul indeksid  $X[i]$  ja  $Y[i]$ .

Aita Aižanil leida kõigis voorudes õiged vahetuskäigud, mille abil ta saaks jada ära järjestada. Mõnes alamülesandes tuleb järjestamist teostada minimaalse võimaliku voorude arvuga. Võib eeldada, et  $M$  vooru on piisav, et jada sorteerida.

Pange tähele, et kui jada saab sorteerituks pärast Ermeki käiku, saab Aižan oma käigul teostada “tühja” vahetuse (näiteks  $0$  ja  $0$ ), ning jada jääb selle vooru lõpuks sorteerituks. Kui jada on juba algselt sorteeritud, on toimuvate vahetusvoorude arvuks  $0$ .

### Näide 1

Olgu:

- Algne jada  $S = 4, 3, 2, 1, 0$ .
- Ermek plaanib osaleda kuni  $M = 6$  voorus.
- Massiivid  $X$  ja  $Y$ , mis kirjeldavad Ermeki poolt valitud indeksite paare, on  $X = 0, 1, 2, 3, 0, 1$  ning  $Y = 1, 2, 3, 4, 1, 2$ . Teisisõnu, Ermek plaanib teostada vahetusi  $(0, 1)$ ,  $(1, 2)$ ,  $(2, 3)$ ,  $(3, 4)$ ,  $(0, 1)$  ja  $(1, 2)$ .

Antud situatsioonis saab Aižan jada  $S$  sorteerida järjekorraks  $0, 1, 2, 3, 4$  kolme vooru abil. Selleks tuleb tal vahetada indeksid  $(0, 4)$ ,  $(1, 3)$  ning  $(3, 4)$ .

Järgnevas tabelis näeme, kuidas Ermek ja Aižan muudavad jada:

Voor	Mängija	Vahetatavad indeksid	Tulemusena saadud jada
algus			4, 3, 2, 1, 0
0	Ermek	(0, 1)	3, 4, 2, 1, 0
0	Aižan	(0, 4)	0, 4, 2, 1, 3
1	Ermek	(1, 2)	0, 2, 4, 1, 3
1	Aižan	(1, 3)	0, 1, 4, 2, 3
2	Ermek	(2, 3)	0, 1, 2, 4, 3
2	Aižan	(3, 4)	0, 1, 2, 3, 4

## Näide 2

Olgu:

- Algne jada  $S = 3, 0, 4, 2, 1$ .
- Ermek plaanib osaleda kuni  $M = 5$  voorus.
- Ermeki poolt planeeritud vahetused on  $(1, 1)$ ,  $(4, 0)$ ,  $(2, 3)$ ,  $(1, 4)$  ja  $(0, 4)$ .

Sel juhul saab Aižan jada  $S$  sorteerida kolme vooruga, näiteks teostades vahetusi  $(1, 4)$ ,  $(4, 2)$  ning  $(2, 2)$ . Järgnevas tabelis näeme, kuidas Ermek ja Aižan muudavad jada:

Voor	Mängija	Vahetatavad indeksid	Tulemusena saadud jada
algus			3, 0, 4, 2, 1
0	Ermek	(1, 1)	3, 0, 4, 2, 1
0	Aižan	(1, 4)	3, 1, 4, 2, 0
1	Ermek	(4, 0)	0, 1, 4, 2, 3
1	Aižan	(4, 2)	0, 1, 3, 2, 4
2	Ermek	(2, 3)	0, 1, 2, 3, 4
2	Aižan	(2, 2)	0, 1, 2, 3, 4

## Ülesanne

Antud on jada  $S$ , arv  $M$  ning massiivid  $X$  ja  $Y$ . Leidke vahetuste järjekord, mille abil Aižan saab jada  $S$  ära sorteerida. Alamülesannetes 5 ja 6 leitud vahetuste järjekord peab olema minimaalne võimalik.

Teil tuleb realiseerida funktsioon `findSwapPairs`:

- `findSwapPairs(N, S, M, X, Y, P, Q)` — Seda funktsiooni kutsutakse välja täpselt ühe korra.
  - $N$ : jada  $S$  pikkus.
  - $S$ : täisarvude massiiv, mis sisaldab algset jada  $S$ .

- $M$ : Ermeki poolt planeeritud vahetuste arv.
- $X, Y$ : täisarvude massiivid pikkusega  $M$ . Iga  $0 \leq i < M$  puhul vahetab Ermek voorus  $i$  jada elemendid indeksitega  $X[i]$  ja  $Y[i]$ .
- $P, Q$ : täisarvude massiivid. Nendes tuleb väljastada üks võimalik vahetuste järjekord, mille abil Aizani õnnestub sorteerida jada  $S$ . Tähistagu  $R$  teie programmi poolt leitud vahetuste arvu. Iga  $0 \leq i < R$  puhul salvestage elementidesse  $P[i]$  ja  $Q[i]$  indeksid, mida Aizan vahetab voorus  $i$ . Võib eeldada, et mõlemale massiividele  $P$  ja  $Q$  on juba eraldatud mälu  $M$  elemendi hoidmiseks.
- Funktsioon peab tagastama väärtuse  $R$  (defineeritud üleval).

## Alamülesanded

alamülesanne	punkte	$N$	$M$	piirangud $X, Y$ -le	nõutud tingimus $R$ -le
1	8	$1 \leq N \leq 5$	$M = N^2$	$X[i] = Y[i] = 0$ iga $i$ korral	$R \leq M$
2	12	$1 \leq N \leq 100$	$M = 30N$	$X[i] = Y[i] = 0$ iga $i$ korral	$R \leq M$
3	16	$1 \leq N \leq 100$	$M = 30N$	$X[i] = 0, Y[i] = 1$ iga $i$ korral	$R \leq M$
4	18	$1 \leq N \leq 500$	$M = 30N$	pole	$R \leq M$
5	20	$6 \leq N \leq 2,000$	$M = 3N$	pole	minimaalne võimalik
6	26	$6 \leq N \leq 200,000$	$M = 3N$	pole	minimaalne võimalik

Võib eeldada, et on olemas lahendus, mis nõuab ülimalt  $M$  vooru.

### Näidishindaja

Näidishindaja loeb sisendit failist `sorting.in` järgnevas formaadis:

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

Näidishindaja trükib välja järgmist väljundit:

- rida 1: funktsiooni `findSwapPairs` väljundväärtus  $R$ ,
- read  $2+i$ , kus  $0 \leq i < R$ :  $P[i] Q[i]$