

# XOR rendezés

Adott egy  $S$  egész szám és egy  $A$  tömb, amiben  $N$  darab nemnegatív egész szám van, 1-től sorszámozva:  $A_1, A_2, \dots, A_N$ . A következő műveletet hajthatjuk végre: egy  $i$  ( $1 \leq i \leq N$ ) sorszámú elemet kiválasztunk:  $A_i$ , majd valamelyik szomszédját, a  $j$ . elemet is kiválasztjuk ( $1 \leq j \leq N$ , ahol  $j = i - 1$  vagy  $j = i + 1$ ). Ezután az  $A_i$  elemet lecseréljük az  $(A_i \oplus A_j)$  értékre, ahol  $\oplus$  a bitenkénti XOR művelet.

A XOR definíciója a leírás végén található.

A cél az  $A$  tömb rendezett tömbbé való alakítása.

- Ha  $S = 1$ , akkor a végső tömbnek szigorúan monoton növekvőnek kell lennie, azaz minden  $i$ -re (ahol  $1 \leq i < N$ ) az  $A_i < A_{i+1}$  kell legyen.
- Ha  $S = 2$ , akkor a végső tömbnek monoton növekvőnek kell lennie, azaz minden  $i$ -re (ahol  $1 \leq i < N$ )  $A_i \leq A_{i+1}$  kell legyen.

Adj meg egy olyan műveletsort, ami eléri a fenti célt.

Nem kell a műveletek számának minimalizálásával foglalkoznod, míg a műveletek száma nem éri el a 40 000-et.

## Bemenet

A bemenet első sora két egész számot tartalmaz:  $N$ -et és  $S$ -et.

A következő sorban  $N$  darab egész szám található, az  $A$  tömb elemei.

## Kimenet

A kimenet első sora egyetlen  $K$  ( $0 \leq K \leq 40000$ ) számot tartalmazzon, a végrehajtandó műveletek számát.

A következő  $K$  sor mindegyike két egészet tartalmazzon, amelyek végrehajtási sorrendben leírják az adott műveleteket: az első egész az  $i$  sorszám, annak az elemnek a sorszáma, amit lecserélünk, a másik egész a  $j$  sorszám: azon elem sorszáma, ami részt vesz a művelet végrehajtásában.

## Korlátok

- $1 \leq S \leq 2$
- $2 \leq N \leq 1000$
- $0 \leq A_i < 2^{20}$

## Részfeladatok

1. (25 pont)  $2 \leq N \leq 150$ ,  $S = 1$ , az  $A$  tömb minden eleme különböző.
2. (35 pont)  $2 \leq N \leq 200$ ,  $S = 1$ , az  $A$  tömb minden eleme különböző.
3. (40 pont)  $2 \leq N \leq 1000$ ,  $S = 2$

## Példák

Bemenet	Kimenet
5 1 3 2 8 4 1	3 1 2 4 3 5 4
5 2 4 4 2 0 1	3 3 2 4 3 5 4

Az első példában a kimenet magyarázata (félkövérrel jelölve az  $i$ . elem):  
[3, 2, 8, 4, 1] -> [1, 2, 8, 4, 1] -> [1, 2, 8, **12**, 1] -> [1, 2, 8, 12, **13**]

A második példában a kimenet magyarázata (félkövérrel jelölve az  $i$ . elem):  
[4, 4, 2, 0, 1] -> [4, 4, **6**, 0, 1] -> [4, 4, 6, **6**, 1] -> [4, 4, 6, 6, **7**]

Ha XOR bitműveletet hajtunk végre az  $x$  és  $y$  biteken, akkor az eredmény 0, ha  $x=y$  és 1 egyébként.

Ha a XOR műveletet hajtjuk végre az  $a$  és  $b$  egészeken, akkor a XOR eredménye a XOR bitművelet végrehajtása az  $a$  és  $b$  minden, egymásnak megfelelő sorszámú bitjére:

$$75 \oplus 29 = 86$$

$$1001011 \oplus 0011101 = 1010110$$

**A C/C++/Java programozási nyelvben a  $\wedge$  operátor (magyar billentyűzetten AltGr+3 karakter) hajtja végre a XOR műveletet.**