

XOR сортирање

Даден ви е цел број S и низа A која има N не-негативни цели броеви, со индекси од 1 натаму. Дозволено ви е да ја извршувате следната операција: одберете го кој било индекс i ($1 \leq i \leq N$), одберете еден од неговите соседи j ($1 \leq j \leq N$, или $j = i - 1$ или $j = i + 1$) и заменете го A_i со $(A_i \oplus A_j)$ каде \oplus е XOR операција по битови. Дефиницијата на XOR ја има на крај на текстот од задачата.

Целта ви е да ја трансформирате A во сортирана низа:

- Ако $S = 1$ тогаш добиената низа мора да е строго растечка, т.е. $A_i < A_{i+1}$ за $1 \leq i < N$
- Ако $S = 2$ тогаш добиената низа мора да е неопадачка, т.е. $A_i \leq A_{i+1}$ за $1 \leq i < N$

Најдете некоја секвенца од операции кои ќе ја постигнат целта.

Не е потребно да го минимизирате бројот на операции, само тој не треба да надмине 40000.

Влез

Во првиот ред има два цели броја: N и S

Во следниот ред има N цели броеви: елементите на A

Излез

Во првиот ред од излезот отпечатете еден цел број K ($0 \leq K \leq 40000$) – бројот на операции.

Во следните K редови отпечатете по 2 цели броја кои ги опишуваат операциите по редослед на извршување: првиот цел број е индекс i на елементот кој ќе се замени, а вториот е индекс j од другиот елемент кој е вклучен во операцијата.

Ограничувања

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

Subtasks

1. (25 поени) $2 \leq N \leq 150$, $S = 1$, Сите елементи во A се различни
2. (35 поени) $2 \leq N \leq 200$, $S = 1$, Сите елементи во A се различни
3. (40 поени) $2 \leq N \leq 1000$, $S = 2$

Примери

Влез	Излез
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

Објаснување на првиот пример:

[3, 2, 8, 4, 1] -> [1, 2, 8, 4, 1] -> [1, 2, 8, **12**, 1] -> [1, 2, 8, 12, **13**]

Објаснување на вториот пример:

[4, 4, 2, 0, 1] -> [4, 4, **6**, 0, 1] -> [4, 4, 6, **6**, 1] -> [4, 4, 6, 6, **7**]

Кога се извршува XOR операција помеѓу битовите а и b резултатот ќе биде 0 ако а=b и 1 во спротивно.

Кога се извршува XOR **операција по битови** помеѓу целите броеви а и b, XOR ќе се направи на секој бит соодветно:

$$75 \oplus 29 = 86$$

$$1001011 \oplus 0011101 = 1010110$$

Во C/C++/Java може да го користите операторот “^” за да извршите XOR операција.