

XOR Sıralama

Sizə bir S tam ədədi və N sayda mənfi-olmayan tam ədədlərdən ibarət A massivi verilir (Massiv 1-dən nömrələnib). Siz massiv üzərində növbəti əməliyyatı icra etməlisiz: hər hansı i ($1 \leq i \leq N$) indeksini seçirsiniz, və onun bir qonşusu olan j ($1 \leq j \leq N$, $j = i - 1$, ya da $j = i + 1$) və A_i elementini ($A_i \oplus A_j$) ilə əvəz edirsiniz (burada \oplus XOR bit əməliyyatını bildirir). XOR əməliyyatının tərifi məsələnin sonunda verilib.

Sizin hədəfiniz A massivinin sıralı vəziyyətə gətirməkdir:

- əgər $S = 1$ olarsa, onda yekun massiv ciddi artan olmalıdır. $A_i < A_{i+1}$, $1 \leq i < N$
- əgər $S = 2$ olarsa, onda massiv azalmayan olmalıdır. $A_i \leq A_{i+1}$, $1 \leq i < N$

Bu hədəfə çatmaq üçün hər hansı əməliyyatlar ardıcılığını tapın.

Maksimal əməliyyat sayı 40000-dən çox olmamalıdır.

Giriş

İlk sətirdə N və S tam ədədlər verilir.

Növbəti sətirdə A massivinin N sayda elementləri verilir.

ÇIXIŞ

İlk sətirdə əməliyyatların sayını bildiren K tam ədədini çapa verməlisiz ($0 \leq K \leq 40000$). Növbəti K sətirin hər bir sətirində iki tam ədəd verilir, bunlar aparılan əməliyyatların xronoloji ardıcılığını göstərir. İlk ədəd əvəz olunan elementin i indeksini və ikinci ədəd bu əməliyyatda iştirak edən qonşusu olan j indeksini bildirir.

Məhdudiyyətlər

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

Alt-tapşırıqlar

1. (25 bal) $2 \leq N \leq 150$, $S = 1$, A massivinin bütün elementləri fərqlidir.
2. (35 bal) $2 \leq N \leq 200$, $S = 1$, A massivinin bütün elementləri fərqlidir.
3. (40 bal) $2 \leq N \leq 1000$, $S = 2$

Nümunələr

Giriş	Çıxış
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

Birinci nümunənin izahı:

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

İkinci nümunənin izahı:

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

a və b bitləri arasında XOR əməliyyatı icra edərkən, əgər $a = b$ olarsa nəticə 0, əks halda nəticə 1 olur.

a və b tam ədədləri arasında XOR əməliyyatı icra edərkən, XOR nəticəsi hər uyğun bitlər üçün icra olunur.

$$75 \oplus 29 = 86$$

$$1001011 \oplus 0011101 = 1010110$$

C/C++/Java dillərində “^” operatoru XOR əməliyyatını icra edir.