

XOR Sıralama

Size bir tamsayı S ve N tane negatif olmayan sayılar içeren bir A dizisi veriliyor. A dizisi 1'den başlayarak indeksleniyor. Bu dizi üzerinde şu işlemi yapabiliyorsunuz: herhangi bir indeks i ($1 \leq i \leq N$) seçip, bir j ($1 \leq j \leq N$, $j = i - 1$ ya da $j = i + 1$) komşusunu seçip A_i değerini ($A_i \oplus A_j$) ile değiştirebilirsiniz. Burada \oplus bitwise XOR operasyonunu göstermektedir. XOR'un nasıl çalıştığını bu dokümanın sonunda görebilirsiniz.

Hedefiniz A dizisini sıralı bir dizi yapmaktır:

- $S = 1$ ise dizinin son hali sürekli artan bir dizi olmalıdır, yani $A_i < A_{i+1}$, $1 \leq i < N$
- $S = 2$ ise dizinin son hali azalmayan bir dizi olmalıdır, yani $A_i \leq A_{i+1}$, $1 \leq i < N$

Hedefinize ulaşmanızı sağlayacak herhangi bir işlem dizisi bulunuz.

İşlem sayısını minimize etmeniz gerekmemektedir. 40000 işlemi aşmamanız yeterlidir.

Girdi

İlk satır iki tamsayı içerir: N ve S

Bir sonraki satır N tamsayı içerir: A dizisinin elemanları

Çıktı

Çıktının ilk satırı tek bir tamsayı içermelidir K ($0 \leq K \leq 40000$) – gerçekleştirilen işlem sayısı.

Sonraki K satır işlemlerin yapılış sırasına göre işlemleri göstermeli ve her birinde iki tamsayı olmalıdır: ilk tamsayı i indeksinde değiştirilecek olan dizi elemanını ve ikincisi de j indeksinde ilgili işlemde kullanılacak diğer elemanı göstermektedir.

Kısıtlar

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

Altgörevler

1. (25 puan) $2 \leq N \leq 150$, $S = 1$, A'nın bütün elemanları birbirinden farklıdır
2. (35 puan) $2 \leq N \leq 200$, $S = 1$, A'nın bütün elemanları birbirinden farklıdır
3. (40 puan) $2 \leq N \leq 1000$, $S = 2$

Örnekler

Girdi	Çıktı
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

İlk örneğin açıklaması:

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

İkinci örneğin açıklaması:

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

a ve b bitleri arasında XOR işlemini gerçekleştirdiğimizde, a b'ye eşitse sonuç 0 olur, a b'den farklıysa 1 olur

a ve b tamsayıları arasında XOR işlemi gerçekleştirdiğimizde, XOR işlemi sayıların karşılıklı bitleri üzerinde tek tek gerçekleştirilerek sonuç elde edilir:

$$75 \oplus 29 = 86$$

$$1001011 \oplus 0011101 = 1010110$$

C/C++/Java'da “^” operatörünü kullanarak XOR işlemi gerçekleştirebilirsiniz.