

# XORanges

Jānim patīk apelsīni, tāpēc viņš izveidoja skeneri ar 4 kamerām un Raspberry Pi un sāka veidot apelsīnu 3D attēlus. Viņa attēlu apstrādes procesors nav pārāk labs - tas izvada tikai 32 bitu skaitli, kas satur informāciju par caurumiem apelsīnu mizā. 32 bitu skaitlis  $D$  tiek attēlots kā virkne 32 ciparu (bitu) virkne. Bitu vērtības var būt 1 vai 0. Ja mēs sākam no 0, tad varam iegūt  $D$  saskaitot  $2^i$  katram  $i$ -tajam bitam, kura vērtība ir 1. Vēl formālāk, skaitlis  $D$  ir attēlots kā virkne  $d_{31}, d_{30}, \dots, d_0$ , kad  $D = d_{31} \cdot 2^{31} + d_{30} \cdot 2^{30} + \dots + d_1 \cdot 2^1 + d_0 \cdot 2^0$ . Piemēram, 13 tiek attēlots kā  $0, \dots, 0, 1, 1, 0, 1$ .

Jānis noskenēja  $n$  apelsīnus. Taču ik pa laikam programmas izpildes laikā viņš nolēma pārskenēt kādu no apelsīniem. Tas nozīmē, ka no pārskenešanās brīža, viņš  $i$ -tajam apelsīnam izmanto jauno vērtību.

Jānis vēlas veikt apelsīnu analīzi. Viņam patīk "izslēdzošā vai" (XOR) darbība, tāpēc Jānis nolēmj veikt noteiktus aprēķinus. Viņš izvēlas noteiktu pēc kārtas esošu apelsīnu intervālu no  $l$  līdz  $u$  (kur  $l \leq u$ ) un vēlas atrast XOR vērtību visām šajā intervālā esošajām vērtībām, visu secīgu elementu pāru vērtībām, visām 3 secīgu elementu virknēm, un tā tālāk līdz pat  $u - l + 1$  secīgu elementu virknei (tas ir, visiem elementiem izvēlētajā intervālā).

Piemēram, ja  $l = 2$  un  $u = 4$ , un ir dots masīvs  $A$  ar visām noskenētajām vērtībām, tad programmai jāatgriež vērtība  $a_2 \oplus a_3 \oplus a_4 \oplus (a_2 \oplus a_3) \oplus (a_3 \oplus a_4) \oplus (a_2 \oplus a_3 \oplus a_4)$ , kur  $\oplus$  ir XOR un  $a_i$  ir  $i$ -tais masīva  $A$  elements.

XOR darbība ir definēta kā:

Ja pirmās vērtības  $i$ -tais bits ir tāds pats kā otrās vērtības  $i$ -tais bits, tad rezultāta  $i$ -tais bits ir 0. Ja pirmās vērtības  $i$ -tais bits ir atšķirīgs no otrās vērtības  $i$ -tā bita, tad rezultāta  $i$ -tais bits ir 1.

$x$	$y$	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Piemēram,  $13 \oplus 23 = 26$ .

13 =	0...001101
------	------------

$23 =$	$0 \dots 010111$
$13 \oplus 23 = 26 =$	$0 \dots 011010$

## Ievaddatu raksturojums

Pirmajā rindā ir doti 2 pozitīvi veseli skaitļi  $n$  un  $q$  (kopējais pārskenējumu un analīžu daudzums - darbību daudzums).

Nākamajā rindā ir doti  $n$  nenegatīvi veseli skaitļi - masīva  $A$  vērtības (sākotnējie apelsīnu skenēšanas rezultāti). Elements  $a_i$  satur  $i$ -tā apelsīna vērtību. Indekss  $i$  sākumā ir 1.

Darbības tiek aprakstītas nākamajās  $q$  rindās ar 3 pozitīviem veseliem skaitļiem katrā rindā.

Ja darbības tips ir 1 (pārskenēšana), tad pirmais skaitlis ir 1, tālāk seko  $i$  (apelsīna indekss, kuru Jānis pārskenēja) un  $j$  ( $i$ -tā apelsīna pārskenēšanas rezultāts)

Ja darbības tips ir 2 (analīzes vaicājums), pirmais skaitlis ir 2, tālāk seko  $l$  un  $u$ .

## Izvaddatu raksturojums

Katram analīzes vaicājuma jāizvada viens skaitlis - šī vaicājuma rezultāts. Katra vērtība jāizvada jaunā rindā.  $i$ -tajā izvades rindā ir jābūt  $i$ -tās analīzes rezultātam. Tev atbildes ir jāizvada tikai 2. tipa darbībām (analīzes darbībām).

## Ierobežojumi

- $a_i \leq 10^9$
- $0 < n, q \leq 2 \cdot 10^5$

## Apakšuzdevumi

1. **[12 punkti]**:  $0 < n, q \leq 100$
2. **[18 punkti]**:  $0 < n, q \leq 500$  un nav pārskenēšanu (1. tipa darbību)
3. **[25 punkti]**:  $0 < n, q \leq 5000$
4. **[20 punkti]**:  $0 < n, q \leq 2 \cdot 10^5$  un nav pārskenēšanu (1. tipa darbību)
5. **[25 punkti]**: Bez ierobežojumiem.

## Piemēri

### 1. piemērs

#### Ievaddati

```
3 3
1 2 3
2 1 3
1 1 3
2 1 3
```

### Izvaddati

```
2
0
```

### Komentāri

Sākumā  $A = [1, 2, 3]$ . Pirmais vaicājums ir par visu intervālu. Analīzes rezultāts ir  $1 \oplus 2 \oplus 3 \oplus (1 \oplus 2) \oplus (2 \oplus 3) \oplus (1 \oplus 2 \oplus 3) = 2$ .

Tad pirmā apelsīna vērtība tiek izmainīta uz 3. Tas izmaina tā paša vaicājuma (intervālā  $[1, 3]$ ) rezultātu  $3 \oplus 2 \oplus 3 \oplus (3 \oplus 2) \oplus (2 \oplus 3) \oplus (3 \oplus 2 \oplus 3) = 0$ .

## 2. piemērs

### Ievaddati

```
5 6
1 2 3 4 5
2 1 3
1 1 3
2 1 5
2 4 4
1 1 1
2 4 4
```

### Izvaddati

```
2
5
4
4
```