

XORanges

Ο Γιάννης αγαπάει τα πορτοκάλια! Έτσι έφτιαξε ένα scanner για πορτοκάλια. Με μια κάμερα και ένα Raspberry Pi 3b+, ξεκίνησε να φτιάχνει 3D εικόνες πορτοκαλιών. Ο επεξεργαστής εικόνας δεν είναι πολύ καλός και έτσι η μόνη έξοδος που παίρνει είναι ένας 32-bit ακέραιος, που περιέχει πληροφορία για τις τρύπες πάνω στην φλούδα του. Ένας 32-bit ακέραιος D αναπαρίσταται ως μία ακολουθία από 32 δυαδικά ψηφία (bits) καθένα από τα οποία είναι είτε 1 είτε 0. Αν ξεκινήσουμε από το 0 μπορούμε να ανακτήσουμε το D προσθέτωντας 2^i για κάθε i -οστό bit που είναι ίσο με το 1. Πιο συγκεκριμένα, ο αριθμός D αναπαρίσταται από την ακολουθία $d_{31}, d_{30}, \dots, d_0$ όταν $D = d_{31} \cdot 2^{31} + d_{30} \cdot 2^{30} + \dots + d_1 \cdot 2^1 + d_0 \cdot 2^0$. Για παράδειγμα, το 13 αναπαρίσταται ως $0, \dots, 0, 1, 1, 0, 1$.

Ο Γιάννης σκανάρει n πορτοκάλια, όμως, κάποιες φορές αποφασίζει να ξανασκανάρει (rescan) κάποιο πορτοκάλι (i -οστό) κατά την εκτέλεση του προγράμματος του. Αυτό σημαίνει ότι από αυτό το σκανάρισμα και μετά, χρησιμοποιεί την ανανεωμένη τιμή του i -οστού πορτοκαλιού.

Ο Γιάννης θέλει να αναλύσει αυτά τα πορτοκάλια. Θεωρεί ότι το αποκλειστικό Ή (XOR) είναι πολύ ενδιαφέρον, γι' αυτό αποφασίζει να κάνει κάποιους υπολογισμούς (queries). Επιλέγει ένα πλήθος (εύρος) πορτοκαλιών από l έως u (όπου $l \leq u$) και θέλει να υπολογίζει την αξία του XOR όλων των στοιχείων σε αυτό το εύρος, όλων των ακολουθιών από 2 διαδοχικά στοιχεία, όλων των ακολουθιών από 3 διαδοχικά στοιχεία ... μέχρι και την ακολουθία από $u - l + 1$ διαδοχικών στοιχείων (όλα τα στοιχεία σε αυτό το εύρος).

Π.χ. αν $l = 2$ και $u = 4$ και υπάρχει ένας πίνακας από σκαναρισμένες τιμές A , το πρόγραμμα θα πρέπει να επιστρέφει την τιμή του $a_2 \oplus a_3 \oplus a_4 \oplus (a_2 \oplus a_3) \oplus (a_3 \oplus a_4) \oplus (a_2 \oplus a_3 \oplus a_4)$, όπου \oplus αναπαριστά το αποκλειστικό Ή (XOR) και a_i είναι το i -οστό στοιχείο στο πίνακα A .

Το αποκλειστικό Ή (XOR) ορίζεται ως ακολούθως:

Αν το i -οστό bit της πρώτης τιμής είναι ίδιο με το i -οστό bit της δεύτερης τιμής, το i -οστό bit του αποτελέσματος της πράξης XOR είναι 0. Αν το i -οστό bit της πρώτης τιμής είναι διαφορετικό με το i -οστό bit της δεύτερης τιμής, το i -οστό bit του αποτελέσματος της πράξης XOR είναι 1.

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

Για παράδειγμα, $13 \oplus 23 = 26$.

$13 =$	$0 \dots 001101$
$23 =$	$0 \dots 010111$
$13 \oplus 23 = 26 =$	$0 \dots 011010$

Είσοδος

Στην πρώτη γραμμή της εισόδου υπάρχουν 2 θετικοί ακέραιοι n και q (όπου q είναι ο συνολικός αριθμός των rescans και queries μαζί, που τα ονομάζουμε *actions*).

Στην επόμενη γραμμή, υπάρχουν n μη-αρνητικοί ακέραιοι (διαχωρισμένοι με κενά), που αντιπροσωπεύουν τις τιμές του πίνακα A . Το a_i περιέχει την τιμή του i -οστού πορτοκαλιού. Ο δείκτης i ξεκινάει από το 1.

Τα *actions* περιγράφονται στις επόμενες q γραμμές με 3 θετικούς ακέραιους διαχωρισμένους με κενά.

Αν ο τύπος του action είναι 1 (rescan), τότε ο πρώτος ακέραιος είναι 1 και ακολουθείται από το i (τον δείκτη του πορτοκαλιού που ξανασκανάρει) και το j (το αποτέλεσμα αυτού του σκαναρίσματος).

Αν ο τύπος του action είναι 2 (query), τότε ο πρώτος ακέραιος είναι 2 και ακολουθείται από το εύρος δηλαδή τους αριθμούς l και u .

Έξοδος

Ένας ακέραιος για κάθε query(action τύπου 2). Θα πρέπει να τυπώνετε κάθε τιμή σε νέα γραμμή. Σημειώστε ότι η i -οστή γραμμή της εξόδου πρέπει να αντιστοιχεί με το αποτέλεσμα του i -οστού query.

Περιορισμοί

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

Υποπροβλήματα

1. **[12 points]**: $0 < n, q \leq 100$
2. **[18 points]**: $0 < n, q \leq 500$ και δεν υπάρχουν actions του τύπου 1
3. **[25 points]**: $0 < n, q \leq 5000$
4. **[20 points]**: $0 < n, q \leq 2 \cdot 10^5$ και δεν υπάρχουν actions του τύπου 1
5. **[25 points]**: Κανένας επιπλέον περιορισμός.

Παραδείγματα

Παράδειγμα 1

Είσοδος

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

Έξοδος

```
2
0
```

Σχόλιο

Στην αρχή δίνεται ο πίνακας $A = [1, 2, 3]$. Το πρώτο query είναι σε όλο το εύρος του πίνακα. Το αποτέλεσμα της ανάλυσης είναι $1 \oplus 2 \oplus 3 \oplus (1 \oplus 2) \oplus (2 \oplus 3) \oplus (1 \oplus 2 \oplus 3) = 2$.

Στην συνέχεια, με το action τύπου 1, η τιμή του 1ου πορτοκαλιού αντικαθίσταται με την τιμή 3.

Αυτό οδηγεί σε μία αλλαγή, καθώς στην συνέχεια εκτελώντας το ίδιο query στο εύρος $[1, 3]$ παίρνουμε διαφορετική τιμή, δηλαδή $3 \oplus 2 \oplus 3 \oplus (3 \oplus 2) \oplus (2 \oplus 3) \oplus (3 \oplus 2 \oplus 3) = 0$.

Παραδειγμα 2

Είσοδος

```
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
```

Έξοδος

2
5
4
4