

XORting

Vremensko ograničenje: 2 sekunde

Memorijsko ograničenje: 1024 MiB

Ovo je interaktivni zadatak.

Iliyan je smjestio Jonasu: sakrio je permutaciju $p_0, p_1, \dots, p_{N-1}$ cijelih brojeva od 0 do $N - 1$. Jonas očajnički želi tu permutaciju otkriti, a ti pristao pomoći mu.

Iliyan neće izravno otkriti permutaciju, ali pristao je odgovarati na pitanja sljedeće vrste: ti odabereš dva indeksa i i j takva da je $0 \leq i, j < N$, a on ti kaže bitovni XOR (ekskluzivno ILI) vrijednosti p_i i p_j .

Bitovni XOR (ekskluzivno ILI) dvaju brojeva, koji se označava s $\oplus$, provodi logičku operaciju ekskluzivnog ILI nad svakim parom odgovarajućih bitova. Rezultat na svakoj poziciji je 1 ako je točno jedan od bitova 1, a bit će 0 ako su oba bita 0 ili oba 1. Na primjer, $75 \oplus 29 = 86$, jer je $1001011_{(2)} \oplus 0011101_{(2)} = 1010110_{(2)}$.

U jeziku C++, XOR vrijednosti a i b možete izračunati pomoću operatora $\wedge$.

Iliyan je spreman odgovoriti na **najviše** Q pitanja.

Međutim, ta pitanja ne mogu uvijek jednoznačno odrediti skrivenu permutaciju. Kažemo da je permutacija $a_0, a_1, \dots, a_{N-1}$ *nerazlučiva* od p ako vrijedi $a_i \oplus a_j = p_i \oplus p_j$ za svaki par indeksa i i j takvih da je $0 \leq i, j < N$. Svaki odgovor koji Iliyan da isti je bez obzira je li skrivena permutacija p ili a , pa nijedan niz pitanja ne može razlikovati te dvije permutacije. Primijetite da je p uvijek nerazlučiva od same sebe, te da možda ne postoji nijedna druga takva permutacija.

Tvoj je zadatak pronaći leksikografski najmanju permutaciju koja je nerazlučiva od p .

Permutacija $x_0, x_1, \dots, x_{N-1}$ leksikografski je manja od permutacije $y_0, y_1, \dots, y_{N-1}$ ako na prvoj poziciji k na kojoj se razlikuju vrijedi $x_k < y_k$. Na primjer, $x = [2, 0, 1, 5, 3]$ leksikografski je manji od $y = [2, 0, 3, 1, 5]$, jer je prva pozicija na kojoj se razlikuju $k = 2$, a $x_2 = 1 < 3 = y_2$.

Skrivena permutacija fiksna je prije nego što tvoj program počne s radom i ne mijenja se kao odgovor na tvoja pitanja.

Detalji implementacije

Vaš program mora implementirati funkciju `solve`:

```
std::vector<int> solve(int N)
```

- N : broj elemenata permutacije.

Ovu funkciju program ocjenjivača poziva točno T puta, jednom za svaku skrivenu permutaciju. Funkcija mora vratiti leksikografski najmanju permutaciju koja je nerazlučiva od permutacije skrivene za taj poziv.

Za komunikaciju s programom ocjenjivača na raspolaganju vam je sljedeća funkcija:

```
int get_xor(int i, int j)
```

Svaki put kad se pozove funkcija `get_xor`, ona vraća $p_i \oplus p_j$. Oba argumenta moraju biti valjani indeksi permutacije. Ako taj uvjet nije zadovoljen, vaše će rješenje dobiti presudu `Output isn't correct: Invalid argument`. Možete pretpostaviti da funkcija odgovara u konstantnom vremenu $O(1)$.

Vrijednost Q fiksna je za svaki podzadatak i ne proslijeđuje se vašem programu.

Ograničenja

- $1 \leq N_i \leq 2^{20}$ za svaki $0 \leq i < T$, gdje je N_i vrijednost od N u i -tom pozivu funkcije `solve`
- $1 \leq T \leq 2^{10}$
- $T \cdot \max(N_0, N_1, \dots, N_{T-1}) \leq 2^{25}$.
- Tijekom i -tog poziva funkcije `solve` u testu smijete postaviti najviše Q pitanja, gdje je Q jednak ili N_i ili N_i^2 , ovisno o podzadatku.

Primjer

Razmotrimo sljedeću interakciju, u kojoj se `solve` poziva dvaput:

Program natjecatelja	Program ocjenjivača
	<code>solve(4)</code>
<code>get_xor(0, 1)</code>	2
<code>get_xor(0, 2)</code>	3
<code>get_xor(0, 3)</code>	1
<code>get_xor(1, 2)</code>	1
<code>get_xor(1, 3)</code>	3
<code>get_xor(2, 3)</code>	2
<code>return {0, 2, 3, 1}</code>	
	<code>solve(1)</code>
<code>return {0}</code>	

Objašnjenje prvog poziva

Skrivena permutacija je $[2, 0, 1, 3]$, no leksikografski najmanja permutacija nerazlučiva od nje je $[0, 2, 3, 1]$. Postavljeno je šest pitanja za $N_1 = 4$, što je dopušteno jer je u ovom podzadatku $Q = N_1^2 = 16$.

Objašnjenje drugog poziva

Jedina moguća permutacija za $N_2 = 1$ je $[0]$.

Podzadaci

Podzadatak	Bodovi	N	T	Q	Dodatna ograničenja
0	0	-	-	N^2	Primjer.
1	7	$\leq 2^3$	2^{10}	N^2	-
2	18	$\leq 2^7$	2^5	N^2	-
3	5	$\leq 2^{11}$	2^5	N^2	-
4	5	$\leq 2^{11}$	2^5	N	-
5	5	$\leq 2^{18}$	2^5	N	$N = 2^k$ za neki cijeli broj k .
6	5	$\leq 2^{18}$	2^5	N	N je neparan.
7	30	$\leq 2^{18}$	2^5	N	-
8	25	$\leq 2^{20}$	2^5	N	-

Primjer programa ocjenjivača

Format ulaza je sljedeći:

- redak 1: dva cijela broja – vrijednosti T i Q_{type} , gdje je Q_{type} jednak ili 1 ili 2;
- sljedećih $2T$ redaka opisuje testove:
 - redak $2i$: jedan cijeli broj – vrijednost N za i -ti test;
 - redak $2i + 1$: N cijelih brojeva – permutacija $p_0, p_1, \dots, p_{N-1}$ za i -ti test.

Ako je $Q_{type} = 1$, tada je $Q = N_i$ za i -ti poziv funkcije `solve`; ako je $Q_{type} = 2$, tada je $Q = N_i^2$.

Format izlaza je sljedeći:

- redak i : permutacija koju je vaš program vratio za i -ti test.