

XORiavimas

Laiko limitas: 2 sekundės

Atminties limitas: 1024 MiB

Tai - interaktyvi užduotis.

Ilijanas iškrėtė pokštą Jonui: jis paslėpė sveikųjų skaičių nuo 0 iki $N - 1$ kėlinį* $p_0, p_1, \dots, p_{N-1}$. Jonas labai nori jį atkurti, o tu sutikai jam padėti.

Skaičių nuo 0 iki $N - 1$ kėliniu* vadiname seką, kurioje lygiai vieną kartą pasikartoja visi skaičiai nuo 0 iki $N - 1$ sumaišyta tvarka.

Ilijanas tiesiogiai neatskleis kėlinio, bet jis sutiko atsakyti į tokio pobūdžio klausimus: gali pasirinkti du indeksus i ir j , kad $0 \leq i, j < N$, ir jis pasakys tau bitinį XOR (griežtas OR) reikšmę tarp p_i ir p_j .

Dviejų skaičių bitinis XOR (griežtas OR), žymimas $\oplus$, atlieka loginę griežtą OR operaciją su kiekviena atitinkamų bitų pora. Rezultatas kiekvienoje pozicijoje yra 1, jei lygiai vienas iš bitų yra 1, bet bus 0, jei abu yra 0 arba abu yra 1. Pavyzdžiui, $75 \oplus 29 = 86$, nes $1001011_{(2)} \oplus 0011101_{(2)} = 1010110_{(2)}$.

C++ kalboje a ir b XOR galite apskaičiuoti naudodami operatorių $\wedge$.

Ilijanas sutinka atsakyti **daugiausiai** į Q klausimų.

Tačiau šie klausimai ne visada leidžia vienareikšmiškai nustatyti paslėptą kėlinį. Sakome, kad kėlinys $a_0, a_1, \dots, a_{N-1}$ yra *neatskiriamas* nuo p , jei $a_i \oplus a_j = p_i \oplus p_j$ kiekvienai indeksų porai i ir j , tokiai, kad $0 \leq i, j < N$. Kiekvienas Ilijano pateiktas atsakymas yra vienodas, nepriklausomai nuo to, ar paslėptas kėlinys yra p , ar a , todėl jokia klausimų seka negali jų atskirti. Atkreipkite dėmesį, kad p visada yra neatskiriamas nuo savęs paties, ir kad gali nebūti jokio kito neatskiriama kėlinio.

Jūsų užduotis – rasti leksikografiškai mažiausią kėlinį, neatskiriamą nuo p .

Kėlinys $x_0, x_1, \dots, x_{N-1}$ yra leksikografiškai mažesnis už $y_0, y_1, \dots, y_{N-1}$, jeigu pirmoje pozicijoje k , kurioje jie skiriasi, galioja $x_k < y_k$. Pavyzdžiui, $x = [2, 0, 1, 4, 3]$ yra leksikografiškai mažesnis už $y = [2, 0, 3, 1, 4]$, nes pirmoji pozicija, kurioje jie skiriasi, yra $k = 2$, ir $x_2 = 1 < 3 = y_2$.

Paslėptas kėlinys yra nustatytas prieš pradedant vykdyti tavo programą ir nesikeičia priklausomai nuo tavo klausimų.

Implementacijos detalės

Turite realizuoti šią funkciją:

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

- N : kėlinio elementų skaičius.

Šią funkciją komisijos programa iškviečia lygiai T kartų, po vieną kartą kiekvienam paslėptam kėliniui. Ji turi grąžinti leksikografiškai mažiausią kėlinį, kurio negalima atskirti nuo tam iškvietimui paslėpto kėlinio.

Bendravimui su komisijos programa jums pateikiama funkcija:

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

Kiekvieną kartą, kai `get_xor` yra kviečiama, ji grąžina $p_i \oplus p_j$. Abu argumentai privalo būti validūs permutacijos indeksai. Jei ši sąlygą nėra patenkinta, jūsų sprendimas gaus verdiktą `Output isn't correct: Invalid argument`. Galite priimti, kad ši funkcija trunką konstantinį laiko kiekį, t.y. $O(1)$.

Q reikšmė yra fiksuota kiekvienai dalinei užduočiai, tačiau ji nėra perduota jūsų programai.

Ribojimai

- $1 \leq N_i \leq 2^{20}$ kiekvienam $1 \leq i < T$, kur N_i yra parametro N reikšmė i -ojo funkcijos `solve` kvietimo metu.
- $1 \leq T \leq 2^{10}$
- $T \cdot \max(N_0, N_1, \dots, N_{T-1}) \leq 2^{25}$.
- i -ojo funkcijos `solve` kvietimo metu jūsų programa gali pateikti nedaugiau Q_i užklausų, kur Q_i yra N_i arba N_i^2 , priklausomai nuo dalinės užduoties.

Pavyzdys

Panagrinėkime šį pavyzdį, kur `solve` iškvieistas dukart:

Dalyvio programa	Komisijos programa
	<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>	

Pirmo kvietimo paaiškinimas

Paslėptas kėlinys yra $[2, 0, 1, 3]$, tačiau leksikografiškai mažiausias, kurio negalima atskirti nuo jo, yra $[0, 2, 3, 1]$. Šešios užklauskos buvo pateiktos, kai $N_1 = 4$, kas yra leistina, nes $Q = N_1^2 = 16$ šiai daliai užduočiai.

Antrojo kvietimo paaiškinimas

Vienintelis galimas kėlinys, kai $N_2 = 1$, yra $[0]$.

Dalinės užduotys

Dalinė užduotis	Taškai	N	T	Q	Papildomi ribojimai
0	0	-	-	N^2	Pavyzdys.
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	$N = 2^k$, kur k yra sveikasis skaičius.
6	5	$\leq 2^{18}$	2^5	N	N yra nelyginis.
7	30	$\leq 2^{18}$	2^5	N	-
8	25	$\leq 2^{20}$	2^5	N	-

Pavyzdinis vertintojas

Pradinių duomenų formatas:

- eilutė 1: du sveikieji skaičiai – T ir Q_{type} reikšmės, kur Q_{type} lygus 1 arba 2;
- kitose $2T$ eilutėse aprašomi testai:
 - eilutėje $2i$: vienas sveikasis skaičius – N reikšmė i -ajam testui;
 - eilutė $2i + 1$: N sveikųjų skaičių – kėlinys $p_0, p_1, \dots, p_{N-1}$ testui i .

Jei $Q_{type} = 1$, tada $Q = N_i$ bus i -ajam `solve` kvietimui; jei $Q_{type} = 2$, tada $Q = N_i^2$.

Išvesties formatas:

- eilutė i : jūsų programos grąžintas kėlinys i -ajam .