

XORtošana

Laika ierobežojums: 2 sekundes

Atmiņas ierobežojums: 1024 MiB

Šis ir interaktīvs uzdevums.

Ilijans ir izspēlējis joku ar Jonasu: viņš ir paslēpis veselo skaitļu no 0 līdz $N - 1$ permutāciju $p_0, p_1, \dots, p_{N-1}$. Jonass izmisīgi vēlas to atgūt, un jūs esat piekritis viņam palīdzēt.

Ilijans neatklās permutāciju tiešā veidā, taču viņš ir piekritis atbildēt uz šāda veida vaicājumiem: jūs varat izvēlēties tādus divus indeksus i un j , ka $0 \leq i, j < N$, un viņš pateiks jums p_i un p_j bitu izslēdzošo VAI (XOR).

Divu skaitļu bitu izslēdzošais VAI (XOR), ko apzīmē ar $\oplus$, katram atbilstošo bitu pārim veic loģisko izslēdzošā VAI operāciju. Rezultāts katrā pozīcijā ir 1, ja tieši viens no bitiem ir 1, bet 0, ja abi ir 0 vai abi ir 1. Piemēram, $75 \oplus 29 = 86$, jo $1001011_{(2)} \oplus 0011101_{(2)} = 1010110_{(2)}$.

Valodā C++ `a` un `b` XOR var aprēķināt, izmantojot operatoru `^`.

Ilijans ir gatavs atbildēt uz **ne vairāk kā** Q vaicājumiem.

Tomēr ar šiem vaicājumiem ne vienmēr var viennozīmīgi noteikt paslēpto permutāciju. Mēs sakām, ka permutācija $a_0, a_1, \dots, a_{N-1}$ ir *neatšķirama* no p , ja $a_i \oplus a_j = p_i \oplus p_j$ katram indeksu pārim i un j , kur $0 \leq i, j < N$. Katra Ilijana atbilde ir vienāda neatkarīgi no tā, vai paslēptā permutācija ir p vai a , tāpēc neviena vaicājumu secība nespēs tās atšķirt. Ievērojiet, ka p vienmēr ir neatšķirama pati no sevis un ka citas neatšķiramas permutācijas var arī nebūt.

Jūsu uzdevums ir atrast leksikogrāfiski mazāko permutāciju, kas ir neatšķirama no p .

Permutācija $x_0, x_1, \dots, x_{N-1}$ ir leksikogrāfiski mazāka nekā $y_0, y_1, \dots, y_{N-1}$, ja pirmajā pozīcijā k , kurā tās atšķiras, izpildās $x_k < y_k$. Piemēram, $x = [2, 0, 1, 4, 3]$ ir leksikogrāfiski mazāka nekā $y = [2, 0, 3, 1, 4]$, jo pirmā pozīcija, kurā tās atšķiras, ir $k = 2$, un $x_2 = 1 < 3 = y_2$.

Paslēptā permutācija tiek fiksēta pirms jūsu programmas darbības sākuma un nemainās atkarībā no jūsu veiktajiem vaicājumiem.

Implementācijas detaļas

Jūsu programmai jāimplementē šāda funkcija:

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

- N : permutācijas elementu skaits.

Šī funkcija tiek izsaukta tieši T reizes — pa vienai reizei katrai paslēptajai permutācijai. Tai jāatgriež leksikogrāfiski mazākā permutācija, kas ir neatšķirama no attiecīgajā izsaukumā paslēptās permutācijas.

Jūsu risinājums var veikt vaicājumus, izsaucot funkciju:

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

Šī funkcija atgriež $p_i \oplus p_j$.

Abiem argumentiem jābūt derīgiem permutācijas indeksiem. Ja šis nosacījums nav izpildīts, jūsu risinājums saņems verdiktu `Output isn't correct: Invalid argument`. Var pieņemt, ka funkcija atbild konstantā laikā $O(1)$.

Vērtība Q ir fiksēta katram apakšuzdevumam un netiek nodota jūsu programmai.

Ierobežojumi

- $1 \leq N_i \leq 2^{20}$ katram $1 \leq i \leq T$, kur N_i ir N vērtība i -tajā funkcijas `solve` izsaukumā
- $1 \leq T \leq 2^{10}$
- $T \cdot \max(N_1, N_2, \dots, N_T) \leq 2^{25}$.
- Testa i -tajā funkcijas `solve` izsaukumā jūs varat uzdot ne vairāk kā Q_i vaicājumus, kur Q_i ir vai nu N_i , vai N_i^2 , atkarībā no apakšuzdevuma.

Piemērs

Aplūkosim šādu mijiedarbību, kurā `solve` tiek izsaukta divas reizes:

Jūsu programma	Vērtēšanas sistēma
	<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>	

1. izsaukuma izskaidrojums

Paslēptā permutācija ir $[2, 0, 1, 3]$, taču leksikogrāfiski mazākā no tās neatšķiramā permutācija ir $[0, 2, 3, 1]$. Ar $N_1 = 4$ tika uzdoti seši vaicājumi, kas ir atļauts, jo šajā apakšuzdevumā $Q_1 = N_1^2 = 16$.

2. izsaukuma izskaidrojums

Vienīgā iespējamā permutācija ar $N_2 = 1$ ir $[0]$.

Apakšuzdevumi

Apakšuzdevums	Punkti	N_i	T	Q_i	Papildu ierobežojumi
0	0	-	-	N_i^2	Piemērs.
1	7	$\leq 2^3$	2^{10}	N_i^2	-
2	18	$\leq 2^7$	2^5	N_i^2	-
3	5	$\leq 2^{11}$	2^5	N_i^2	-
4	5	$\leq 2^{11}$	2^5	N_i	-
5	5	$\leq 2^{18}$	2^5	N_i	$N_i = 2^k$ kādam veselam skaitlim k .
6	5	$\leq 2^{18}$	2^5	N_i	N_i ir nepāra skaitlis.
7	30	$\leq 2^{18}$	2^5	N_i	-
8	25	$\leq 2^{20}$	2^5	N_i	-

Piemēra vērtētājprogramma

Piemēra vērtētājprogramma nolasa ievadi šādā formātā:

- 1. rinda: divi veseli skaitļi – T un Q_{type} vērtības, kur Q_{type} ir vai nu 1, vai 2;
- nākamās $2T$ rindas apraksta testus:
 - $2i$. rinda: viens vesels skaitlis – N_i vērtība i -tajam testam;
 - $2i + 1$. rinda: N_i veseli skaitļi – permutācija $p_0, p_1, \dots, p_{N_i-1}$ i -tajam testam.

Ja $Q_{type} = 1$, tad i -tajā funkcijas `solve` izsaukumā $Q_i = N_i$; ja $Q_{type} = 2$, tad $Q_i = N_i^2$.

Vērtētājprogramma izvadīs rezultātu šādā formātā:

- i . rinda: permutācija, ko jūsu programma atgriež i -tajā testā.