

XORting

Time limit: 2 seconds
Memory limit: 1024 MiB

Սա ինտերակտիվ խնդիր է:

Հայկը գլուխկոտրուկ է առաջարկել Վաչեին. նա թաքցրել է $p_0, p_1, \dots, p_{N-1}$ 0-ից $N - 1$ թվերի տեղափոխությունը: Վաչեն հուսահատ ցանկանում է այն վերականգնել, և դուք համաձայնվել եք օգնել նրան այդ հարցում:

Հայկը չի ուզում բացահայտել տեղափոխությունը ուղղակիորեն, սակայն նա համաձայն է պատասխանել հետևյալ տեսակի հարցերին. դուք կարող եք ընտրել երկու ինդեքս՝ i և j , այնպես, որ $0 \leq i, j < N$, և նա ասում է ձեզ p_i և p_j թվերի բիթային XOR-ը:

Երկու թվերի բիթային XOR-ը, նշվում է $\oplus$ -ով, կատարում է տրամաբանական բացառող ԿԱՄ գործողություն համապատասխան բիթերի յուրաքանչյուր զույգի վրա: Արդյունքը յուրաքանչյուր դիրքում 1 է, եթե ճիշտ մեկ բիթ է 1, բայց կլինի 0, եթե երկուսն էլ 0 են կամ երկուսն էլ 1 են: Օրինակ, $75 \oplus 29 = 86$, քանի որ $1001011_{(2)} \oplus 0011101_{(2)} = 1010110_{(2)}$:

C++-ում կարելի է հաշվել a և b XOR-ը՝ օգտագործելով $\wedge$ օպերատորը:

Հայկը համաձայն է պատասխանել **առավելագույնը** Q հարցերի:

Սակայն այս հարցերը միշտ չէ, որ կարող են միարժեքորեն որոշել թաքցված տեղափոխությունը: Մենք սահմանում ենք $a_0, a_1, \dots, a_{N-1}$ տեղափոխությունը որպես p -ից **չտարբերելի**, եթե $a_i \oplus a_j = p_i \oplus p_j$ $0 \leq i, j < N$ ինդեքսների յուրաքանչյուր զույգի համար: Հայկը տալիս է միանման պատասխաններ, լինի թաքցված տեղափոխությունը p թե a , այնպես որ հարցերի ոչ մի հաջորդականություն չի կարող դրանք տարբերակել: Նշենք, որ p -ն միշտ չտարբերելի է ինքն իրենից, և դրանից բացի այլ չտարբերելի տեղափոխություն կարող է գոյություն չունենալ:

Ձեր խնդիրն է գտնել p -ից չտարբերելի լեքսիկոգրաֆիկորեն ամենափոքր տեղափոխությունը:

$x_0, x_1, \dots, x_{N-1}$ տեղափոխությունը լեքսիկոգրաֆիկորեն փոքր է $y_0, y_1, \dots, y_{N-1}$ տեղափոխությունից, եթե առաջին այն k դիրքում որտեղ դրանք տարբերվում են, $x_k < y_k$: Օրինակ, $x = [2, 0, 1, 5, 3]$ բառարանային

կարգով փոքր է $y = [2, 0, 3, 1, 5]$ -ից, քանի որ առաջին դիրքը, որտեղ դրանք տարբերվում են, $k = 2$ է, և $x_2 = 1 < 3 = y_2$:

Թաքցված տեղափոխությունը ֆիքսված է ձեր ծրագրի մեկնարկից առաջ և չի փոխվում ձեր հարցերին ի պատասխան:

Իրականացման մանրամասներ

Ձեր ծրագիրը պետք է իրականացնի `solve` ֆունկցիան.

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

- N . տեղափոխության տարրերի քանակը:

Այս ֆունկցիան կանչվում է ուղիղ T անգամ ժյուրիի ծրագրի կողմից, յուրաքանչյուր թաքցված տեղափոխության համար մեկ անգամ: Այն պետք է վերադարձնի լեքսիկոգրաֆիկորեն ամենափոքր տեղափոխությունը, որը չտարբերելի է այդ կանչի համար թաքցված տեղափոխությունից:

Ժյուրիի ծրագրի հետ հաղորդակցվելու համար, տրված է հետևյալ ֆունկցիան.

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

Ամեն անգամ, երբ `get_xor` ֆունկցիան կանչվում է, այն վերադարձնում է $p_i \oplus p_j$: Երկու արգումենտներն էլ պետք է տեղափոխության վավեր ինդեքսներ լինեն: Եթե այս պայմանը չի բավարարվում, ձեր լուծումը կստանա հետևյալ վճիռը `Output isn't correct: Invalid argument:` Դուք կարող եք ենթադրել, որ ֆունկցիան պատասխանում է հաստատուն ժամանակում $O(1)$:

Q արժեքը ամրագրված է յուրաքանչյուր ենթախնդրի համար և չի փոխանցվում ձեր ծրագրին:

Սահմանափակումներ

- $1 \leq N_i \leq 2^{20}$ յուրաքանչյուր $1 \leq i \leq T$ -ի համար, որտեղ N_i i -րդ `solve` կանչի արժեքն է
- $1 \leq T \leq 2^{10}$
- $T \cdot \max(N_1, N_2, \dots, N_T) \leq 2^{25}$:
- `solve` i -րդ կանչի ժամանակ թեստի մեջ կարող եք հարցնել առավելագույնը Q հարց, որտեղ Q N_i կամ N_i^2 է, կախված ենթախնդրից:

Օրինակ

Դիտարկենք հետևյալ ինտերակցիան, որում `solve` ֆունկցիան կանչվում է երկու անգամ.

Մասնակցի ծրագիր	Ժյուրիի ծրագիր
	<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>	

Առաջին կանչի բացատրությունը

Թաքցված տեղափոխությունը $[2, 0, 1, 3]$ է, բայց դրանից չտարբերելի լեքսիկոգրաֆիկորեն ամենափոքր տեղափոխությունը $[0, 2, 3, 1]$ է: $N_1 = 4$ -ի համար տրվեց վեց հարց, ինչը թույլատրվում է, քանի որ այս ենթախնդիրում $Q = N_1^2 = 16$:

Երկրորդ կանչի բացատրությունը

$N_2 = 1$ -ի համար միակ հնարավոր տեղափոխությունը $[0]$ է:

Ենթախնդիրներ

Ենթախնդիր	Միավորներ	N	T	Q	Լրացուցիչ սահմանափակումներ
0	0	-	-	N^2	Օրինակը:
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	$N = 2^k$ ինչ-որ ամբողջ k -ի համար:
6	5	$\leq 2^{18}$	2^5	N	N կենտ է:
7	30	$\leq 2^{18}$	2^5	N	-
8	25	$\leq 2^{20}$	2^5	N	-

Գրեյդերի նմուշ

Մուտքի ձևաչափը հետևյալն է.

- տող 1: երկու ամբողջ թիվ – T և Q_{type} արժեքները, որտեղ Q_{type} -ը կամ 1 է, կամ 2;
- հաջորդ $2T$ տողերը նկարագրում են թեստերը.
 - տող $2i$: մեկ ամբողջ թիվ – i -րդ թեստի N արժեքը;
 - տող $2i + 1$: N ամբողջ թիվ – i -րդ թեստի $p_0, p_1, \dots, p_{N-1}$ տեղափոխությունը:

Եթե $Q_{type} = 1$, ապա solve-ի i -րդ կանչի համար $Q = N_i$; եթե $Q_{type} = 2$, ապա $Q = N_i^2$:

Ելքի ձևաչափը հետևյալն է.

- տող i : ձեր ծրագրի վերադարձրած տեղափոխությունը i -րդ թեստի համար: