НАЦИОНАЛЕН ЕСЕНЕН ТУРНИР ПО ИНФОРМАТИКА „ДЖОН АТАНАСОВ“
София, 27 – 29 ноември 2026 г.
Група А – 11, 12 клас

Задача А1. Битмакс

⏳ 5 сек. 💾 1024 MB

Клод (човекът, не изкуственият интелект) Шанън е открил как ефективно да извършва булеви операции с помощта на електрически вериги. Сега той се чуди как да обработва големи количества числа наведнъж. По-точно, той разполага с множество от N цели неотрицателни 63-битови числа и иска многократно да прилага побитови операции[1] върху всяко от тях (на дадена стъпка една и съща операция се прилага върху всички числа):

- `XOR` с X (“`Y = Y ^ X`” в C++);
- `AND` с X (“`Y = Y & X`” в C++);
- `OR` с X (“`Y = Y | X`” в C++).

След прилагането на всяка операция Шанън иска да знае кое е най-голямото число в множеството. Обърнете внимание, че операциите се натрупват, т.е. всяка операция се прилага върху числата, получени след прилагането на предишните операции.

За съжаление, N е твърде голямо, а Шанън е ограничен от технологиите на своето време. Помогнете му, като напишете програма **`bitmax`**, която решава неговия проблем.

Детайли по имплементацията

Трябва да имплементирате функциите `init` и `apply`:

```
void init(std::vector<long long> A)
```

Тази функция ще бъде извикана веднъж в началото на всеки тест и ще предостави на Вашата програма вектора A с N-те числа.

```
long long apply(int T, long long X)
```

Тази функция ще бъде извикана Q пъти за всеки тест. Всяко извикване съответства на операция, която да бъде приложена върху числата. Тук X е стойността, а T е видът на операцията: 0 за `XOR`, 1 за `AND` и 2 за `OR`. Функцията трябва да върне най-голямото число от множеството след прилагането на операцията.

Ограничения

- $1 \le N, Q \le 6 \times 10^5$;
- $0 \le A_i < 2^{63}$;
- $0 \le X < 2^{63}$.

[1] Побитова операция върху две числа се прилага поотделно за всяка позиция в двоичните им записи: битът на резултата е 1, ако двата бита са различни (`XOR`), ако и двата са 1 (`AND`) или ако поне единият е 1 (`OR`). Например за $12 = 1100_2$ и $10 = 1010_2$ се получава съответно $0110_2 = 6$, $1000_2 = 8$ и $1110_2 = 14$.

Подзадачи

Подзадача	Точки	Необходими подзадачи	Ограничения
0	0	–	Примерът.
1	5	0	$N, Q \leq 10^4$
2	17	–	$T = 1$ или $T = 2$
3	29	–	$T = 0$
4	16	$0 - 1$	$N, Q \leq 1.5 \times 10^5$
5	12	$0 - 1, 4$	$N, Q \leq 4 \times 10^5$
6	21	$0 - 5$	–

Точките за дадена подзадача се получават само ако се преминат успешно всички тестове, предвидени за нея и необходимите подзадачи.

Пример

Вход	Изход
5 19 5 14 29 4 5 0 0 1 15 0 23 0 31 2 20	29 14 26 13 31

Обяснение на примера.

Операция	Числа след операцията
XOR 0	19 5 14 **29** 4
AND 15	3 5 **14** 13 4
XOR 23	20 18 25 **26** 19
XOR 31	11 **13** 6 5 12
OR 20	**31** 29 22 21 28

Локален грейдър

Формат на входа:

- ред 1: едно цяло число N – броят на числата;
- ред 2: N цели неотрицателни числа $A_0\ A_1\ \ldots\ A_{N-1}$ – началните числа;
- ред 3: едно цяло число Q – броят на операциите;
- ред $4 + i$ (за $0 \leq i < Q$): две цели числа $T_i\ X_i$ – видът и стойността на i-тата операция.

Формат на изхода:

- ред $1 + i$ (за $0 \leq i < Q$): едно цяло число – върнатата стойност при i-тото извикване на `apply`.