

Мистический эксперимент Пагсли

Автор задачи: Егор Юлин, разработчик: Иван Пашков

Заметим, что если при массиве a_1, a_2, \dots, a_n для некоторого элемента a_i выполняется, что $\text{xor}(a_1, a_2, \dots, a_{i-1})$ равняется $\text{xor}(a_{i+1}, a_{i+2}, \dots, a_n)$, то суммарный $\text{xor}(a_1, a_2, \dots, a_n)$ массива равняется a_i (потому что xor префикса равняется xor суффикса, а значит их суммарный хог равен 0 по свойству xor).

Таким образом, чтобы вычислить ответ для текущего массива, нужно понять, сколько элементов в массиве в точности равняется хог всего массива.

Для этого будем поддерживать количество элементов для каждого значения с помощью какого-нибудь словаря, например, с помощью `std::unordered_map<int, int>` в C++ или `java.util.HashMap<Integer, Integer>` в Java.

Будем также поддерживать суммарный xor массива. Как пересчитывать? Пусть до замены элемента суммарный xor элементов равняется `sum`, старое значение элемента равняется `old_value`, новое равняется `new_value`. При замене элемента новый суммарный xor массива равняется $\text{sum} \oplus \text{old_value} \oplus \text{new_value}$.

Итоговый алгоритм:

1. В самом начале инициализируем словарь всеми значениями и считаем суммарный xor массива.
2. При запросе первого типа пересчитываем суммарный xor массива и обновляем данные в словаре;
3. При запросе второго типа получаем из словаря и выводим количество значений, которые равняются суммарному xor всех элементов.