Problem 3632: Subarrays with XOR at Least K

Problem Information

Difficulty: Hard
Acceptance Rate: 44.55%
Paid Only: Yes
Tags: Array, Bit Manipulation, Trie, Prefix Sum

Problem Description

Given an array of positive integers `nums` of length `n` and a non■negative integer `k`.

Return the number of **contiguous subarrays** whose bitwise XOR of all elements is **greater** than or **equal** to `k`.

Example 1:

Input: nums = [3,1,2,3], k = 2

Output: 6

Explanation:

The valid subarrays with `XOR >= 2` are `[3]` at index 0, `[3, 1]` at indices 0 - 1, `[3, 1, 2, 3]` at indices 0 - 3, `[1, 2]` at indices 1 - 2, `[2]` at index 2, and `[3]` at index 3; there are 6 in total.

Example 2:

Input: nums = [0,0,0], k = 0

Output: 6

Explanation:

Every contiguous subarray yields `XOR = 0`, which meets `k = 0`. There are 6 such subarrays in total.

Constraints:

* `1 <= nums.length <= 105` * `0 <= nums[i] <= 109` * `0 <= k <= 109`

Code Snippets

C++:

```
class Solution {
public:
long long countXorSubarrays(vector<int>& nums, int k) {

}
};
```

Java:

```
class Solution {
public long countXorSubarrays(int[] nums, int k) {

}
}
```

Python3:

```
class Solution:
def countXorSubarrays(self, nums: List[int], k: int) -> int:
```