Problem 3599: Partition Array to Minimize XOR

Problem Information

Difficulty: Medium
Acceptance Rate: 40.67%
Paid Only: No
Tags: Array, Dynamic Programming, Bit Manipulation, Prefix Sum

Problem Description

You are given an integer array `nums` and an integer `k`.

Your task is to partition `nums` into `k`**** non-empty **subarrays**. For each subarray, compute the bitwise **XOR** of all its elements.

Return the **minimum** possible value of the **maximum XOR** among these `k` subarrays.

Example 1:

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

Output: 1

Explanation:

The optimal partition is `[1]` and `[2, 3]`.

* XOR of the first subarray is `1`. * XOR of the second subarray is `2 XOR 3 = 1`.

The maximum XOR among the subarrays is 1, which is the minimum possible.

Example 2:

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

Output: 2

Explanation:

The optimal partition is `[2]`, `[3, 3]`, and `[2]`.

* XOR of the first subarray is `2`. * XOR of the second subarray is `3 XOR 3 = 0`. * XOR of the third subarray is `2`.

The maximum XOR among the subarrays is 2, which is the minimum possible.

Example 3:

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

Output: 0

Explanation:

The optimal partition is `[1, 1]` and `[2, 3, 1]`.

* XOR of the first subarray is `1 XOR 1 = 0`. * XOR of the second subarray is `2 XOR 3 XOR 1 = 0`.

The maximum XOR among the subarrays is 0, which is the minimum possible.

Constraints:

* `1 <= nums.length <= 250` * `1 <= nums[i] <= 109` * `1 <= k <= n`

Code Snippets

C++:

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

}

};
```

Java:

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

}
}
```

Python3:

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