Problem 3681: Maximum XOR of Subsequences

Problem Information

Difficulty: Hard
Acceptance Rate: 50.96%
Paid Only: No
Tags: Array, Math, Greedy, Bit Manipulation

Problem Description

You are given an integer array `nums` of length `n` where each element is a non-negative integer.

Select **two** subsequences of `nums` (they may be empty and are **allowed** to **overlap**), each preserving the original order of elements, and let:

* `X` be the bitwise XOR of all elements in the first subsequence. * `Y` be the bitwise XOR of all elements in the second subsequence.

Return the **maximum** possible value of `X XOR Y`.

Note: The XOR of an **empty** subsequence is 0.

Example 1:

Input: nums = [1,2,3]

Output: 3

Explanation:

Choose subsequences:

* First subsequence `[2]`, whose XOR is 2. * Second subsequence `[2,3]`, whose XOR is 1.

Then, XOR of both subsequences = `2 XOR 1 = 3`.

This is the maximum XOR value achievable from any two subsequences.

Example 2:

Input: nums = [5,2]

Output: 7

Explanation:

Choose subsequences:

* First subsequence `[5]`, whose XOR is 5. * Second subsequence `[2]`, whose XOR is 2.

Then, XOR of both subsequences = `5 XOR 2 = 7`.

This is the maximum XOR value achievable from any two subsequences.

Constraints:

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

Code Snippets

C++:

```
class Solution {
public:
int maxXorSubsequences(vector<int>& nums) {

}
};
```

Java:

```
class Solution {
public int maxXorSubsequences(int[] nums) {
}
}
```

Python3:

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