

# Problem 3630: Partition Array for Maximum XOR and AND

## Problem Information

**Difficulty:** Hard

**Acceptance Rate:** 13.48%

**Paid Only:** No

**Tags:** Array, Math, Greedy, Bit Manipulation, Enumeration

## Problem Description

You are given an integer array `nums`.

Partition the array into **three** (possibly empty) subsequences `A`, `B`, and `C` such that every element of `nums` belongs to **exactly** one subsequence.

Your goal is to **maximize** the value of:  $\text{XOR}(A) + \text{AND}(B) + \text{XOR}(C)$

where:

\*  $\text{XOR}(\text{arr})$  denotes the bitwise XOR of all elements in `arr`. If `arr` is empty, its value is defined as 0. \*  $\text{AND}(\text{arr})$  denotes the bitwise AND of all elements in `arr`. If `arr` is empty, its value is defined as 0.

Return the **maximum** value achievable.

**Note:** If multiple partitions result in the same **maximum** sum, you can consider any one of them.

**Example 1:**

**Input:** `nums = [2,3]`

**Output:** 5

**\*\*Explanation:\*\***

One optimal partition is:

\* `A = [3], XOR(A) = 3` \* `B = [2], AND(B) = 2` \* `C = [], XOR(C) = 0`

The maximum value of: `XOR(A) + AND(B) + XOR(C) = 3 + 2 + 0 = 5`. Thus, the answer is 5.

**\*\*Example 2:\*\***

**\*\*Input:\*\*** nums = [1,3,2]

**\*\*Output:\*\*** 6

**\*\*Explanation:\*\***

One optimal partition is:

\* `A = [1], XOR(A) = 1` \* `B = [2], AND(B) = 2` \* `C = [3], XOR(C) = 3`

The maximum value of: `XOR(A) + AND(B) + XOR(C) = 1 + 2 + 3 = 6`. Thus, the answer is 6.

**\*\*Example 3:\*\***

**\*\*Input:\*\*** nums = [2,3,6,7]

**\*\*Output:\*\*** 15

**\*\*Explanation:\*\***

One optimal partition is:

\* `A = [7], XOR(A) = 7` \* `B = [2,3], AND(B) = 2` \* `C = [6], XOR(C) = 6`

The maximum value of: `XOR(A) + AND(B) + XOR(C) = 7 + 2 + 6 = 15`. Thus, the answer is 15.

**\*\*Constraints:\*\***

\*`1 <= nums.length <= 19` \*`1 <= nums[i] <= 109`

## Code Snippets

### C++:

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

    }
};
```

### Java:

```
class Solution {
    public long maximizeXorAndXor(int[] nums) {

    }
}
```

### Python3:

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