Problem 1863: Sum of All Subset XOR Totals

Problem Information

Difficulty: Easy
Acceptance Rate: 90.06%
Paid Only: No
Tags: Array, Math, Backtracking, Bit Manipulation, Combinatorics, Enumeration

Problem Description

The **XOR total** of an array is defined as the bitwise `XOR` of**all its elements** , or `0` if the array is**empty**.

* For example, the **XOR total** of the array `[2,5,6]` is `2 XOR 5 XOR 6 = 1`.

Given an array `nums`, return _the**sum** of all **XOR totals** for every **subset** of _`nums`.

Note: Subsets with the **same** elements should be counted **multiple** times.

An array `a` is a **subset** of an array `b` if `a` can be obtained from `b` by deleting some (possibly zero) elements of `b`.

Example 1:

Input: nums = [1,3] **Output:** 6 **Explanation:** The 4 subsets of [1,3] are: - The empty subset has an XOR total of 0. - [1] has an XOR total of 1. - [3] has an XOR total of 3. - [1,3] has an XOR total of 1 XOR 3 = 2. 0 + 1 + 3 + 2 = 6

Example 2:

Input: nums = [5,1,6] **Output:** 28 **Explanation:** The 8 subsets of [5,1,6] are: - The empty subset has an XOR total of 0. - [5] has an XOR total of 5. - [1] has an XOR total of 1. - [6] has an XOR total of 6. - [5,1] has an XOR total of 5 XOR 1 = 4. - [5,6] has an XOR total of 5 XOR 6 = 3. - [1,6] has an XOR total of 1 XOR 6 = 7. - [5,1,6] has an XOR total of 5 XOR 1 XOR 6 = 2. 0 + 5 + 1 + 6 + 4 + 3 + 7 + 2 = 28

Example 3:

Input: nums = [3,4,5,6,7,8] **Output:** 480 **Explanation:** The sum of all XOR totals for every subset is 480.

Constraints:

* `1 <= nums.length <= 12` * `1 <= nums[i] <= 20`

Code Snippets

C++:

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

}
};
```

Java:

```
class Solution {
public int subsetXORSum(int[] nums) {

}
}
```

Python3:

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