Problem 137: Single Number II

Problem Information

Difficulty: Medium
Acceptance Rate: 66.22%
Paid Only: No
Tags: Array, Bit Manipulation

Problem Description

Given an integer array `nums` where every element appears **three times** except for one, which appears **exactly once**. _Find the single element and return it_.

You must implement a solution with a linear runtime complexity and use only constant extra space.

Example 1:

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

Example 2:

Input: nums = [0,1,0,1,0,1,99] **Output:** 99

Constraints:

* `1 <= nums.length <= 3 * 104` * `-231 <= nums[i] <= 231 - 1` * Each element in `nums` appears exactly **three times** except for one element which appears **once**.

Code Snippets

C++:

```
class Solution {
public:
```

```
int singleNumber(vector<int>& nums) {

}
};
```

Java:

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

}
}
```

Python3:

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