Problem 3289: The Two Sneaky Numbers of Digitville

Problem Information

Difficulty: Easy
Acceptance Rate: 89.92%
Paid Only: No
Tags: Array, Hash Table, Math

Problem Description

In the town of Digitville, there was a list of numbers called `nums` containing integers from `0` to `n - 1`. Each number was supposed to appear **exactly once** in the list, however, **two** mischievous numbers sneaked in an _additional time_ , making the list longer than usual.

As the town detective, your task is to find these two sneaky numbers. Return an array of size **two** containing the two numbers (in _any order_), so peace can return to Digitville.

Example 1:

Input: nums = [0,1,1,0]

Output: [0,1]

Explanation:

The numbers 0 and 1 each appear twice in the array.

Example 2:

Input: nums = [0,3,2,1,3,2]

Output: [2,3]

Explanation:

The numbers 2 and 3 each appear twice in the array.

Example 3:

Input: nums = [7,1,5,4,3,4,6,0,9,5,8,2]

Output: [4,5]

Explanation:

The numbers 4 and 5 each appear twice in the array.

Constraints:

* `2 <= n <= 100` * `nums.length == n + 2` * `0 <= nums[i] < n` * The input is generated such that `nums` contains **exactly** two repeated elements.

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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