Problem 645: Set Mismatch

Problem Information

Difficulty: Easy
Acceptance Rate: 45.27%
Paid Only: No
Tags: Array, Hash Table, Bit Manipulation, Sorting

Problem Description

You have a set of integers `s`, which originally contains all the numbers from `1` to `n`. Unfortunately, due to some error, one of the numbers in `s` got duplicated to another number in the set, which results in **repetition of one** number and **loss of another** number.

You are given an integer array `nums` representing the data status of this set after the error.

Find the number that occurs twice and the number that is missing and return _them in the form of an array_.

Example 1:

Input: nums = [1,2,2,4] **Output:** [2,3]

Example 2:

Input: nums = [1,1] **Output:** [1,2]

Constraints:

* `2 <= nums.length <= 104` * `1 <= nums[i] <= 104`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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