442. Find All Duplicates in an Array

Given an integer array `nums` of length `n` where all the integers of `nums` are in the range `[1, n]` and each integer appears **once** or **twice**, return *an array of all the integers that appears **twice***.

You must write an algorithm that runs in `O(n)` time and uses only constant extra space.

Example 1:

Input: nums = [4,3,2,7,8,2,3,1]
Output: [2,3]

Example 2:

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

Example 3:

Input: nums = [1]
Output: []

Constraints:

- `n == nums.length`
- `1 ≤ n ≤ 105`
- `1 ≤ nums[i] ≤ n`
- Each element in `nums` appears **once** or **twice**.

```java
class Solution {
    public List<Integer> findDuplicates(int[] nums) {
        List<Integer> duplicates = new ArrayList<>();
        for (int num : nums) {
            int index = Math.abs(num) - 1;
            if (nums[index] < 0) {
                duplicates.add(Math.abs(num));
            } else {
                nums[index] = -nums[index];
            }
        }
        return duplicates;
```

```
    }
}
```