

[Explore](#)[Problems](#)[Interview](#)

New

[Contest](#)[Discuss](#)[Store](#)[Premium](#)[Description](#)[Solution](#)[Discuss \(999+\)](#)[Submissions](#)[Python3](#)

442. Find All Duplicates in an Array

Medium

4904

223

Add to List

Share

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`

Aut

`i {}`

```
1 class Solution:
2     def findDuplicates(self, nums: List[int]) -> List[int]:
3         #
4         numbers are in the range [1, n]
5         duplicate
6         0:
7
8         duplicate
9
10        * = -1
11        re
```

[Testcase](#) [Run Code F](#)

You have attempted
Please try again in :
[subscribe to reduce](#)

[Console](#) [Use E](#)[Problems](#)[Pick One](#)[Prev](#)

79/336

[Next](#)[Run Code](#)