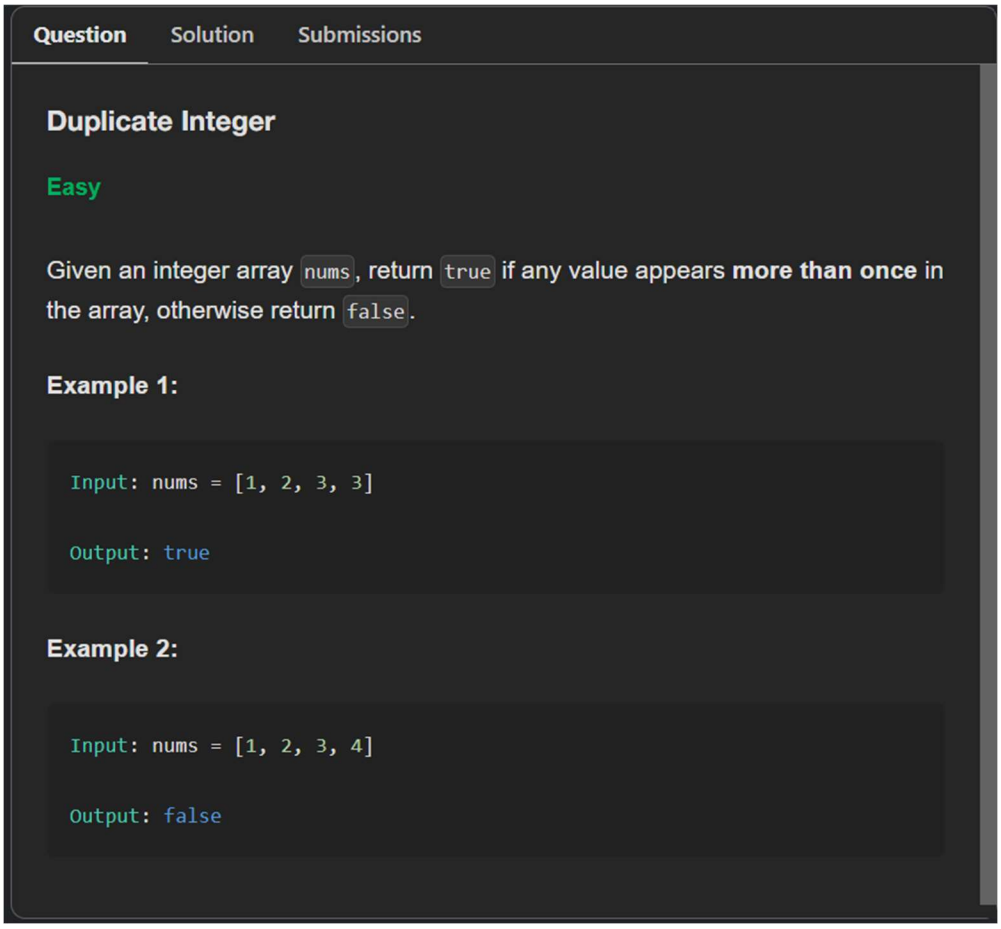
Problem1: Duplicate Integer



Question Solution Submissions

Duplicate Integer

Easy

Given an integer array `nums`, return `true` if any value appears **more than once** in the array, otherwise return `false`.

Example 1:

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

Output: true
```

Example 2:

```
Input: nums = [1, 2, 3, 4]

Output: false
```

Solution1: Brute Force

- Time Complexity: O(n^2)
- Space Complexity: O(1)

```python
# Solution1: BruteForce
def hasDuplicate(self, nums: List[int]) -> bool:
    ## Choose one element
    for i in range(len(nums)):
        ## Iterate from i+1 the element for comparison
        for j in range(i+1, len(nums)):
            # Check duplicates
            if nums[i] == nums[j]:
                return True
    return False
```

Solution2: Sorting

- First, Sort the array and check the if any 2 neighbors are same.
- Time Complexity: O (n*log n) => for sorting array
- Space Complexity: O (1)

```python
def hasDuplicate(self, nums: List[int]) -> bool:
    # Sort the array : It takes nlogn time
    sorted_nums = sorted(nums)
    for i in range(len(sorted_nums)):
        # next element
        j = i+1
        # J<len(nums)  ==> To avoid list index out of error
        # check duplicates
        if j<len(nums) and sorted_nums[i] == sorted_nums[j]:
            return True
    return False
```

Solution3: HashSet

- Use hashset to check if we already visited that number, It required extra space O(n) but it decreases time to O(n)
- Time Complexity: O(n)
- Space Complexity: O(n)

```python
## Solution3 --> In different way
def hasDuplicate(self, nums: List[int]) -> bool:
    # Extra Space : O(n)
    hashset = set()

    for num in nums:
        if num in hashset:
            return True
        hashset.add(num)
    return False
```