Problem 1752: Check if Array Is Sorted and Rotated

Problem Information

Difficulty: Easy
Acceptance Rate: 55.47%
Paid Only: No
Tags: Array

Problem Description

Given an array `nums`, return `true` _if the array was originally sorted in non-decreasing order, then rotated**some** number of positions (including zero)_. Otherwise, return `false`.

There may be **duplicates** in the original array.

Note: An array `A` rotated by `x` positions results in an array `B` of the same length such that `B[i] == A[(i+x) % A.length]` for every valid index `i`.

Example 1:

Input: nums = [3,4,5,1,2] **Output:** true **Explanation:** [1,2,3,4,5] is the original sorted array. You can rotate the array by x = 2 positions to begin on the element of value 3: [3,4,5,1,2].

Example 2:

Input: nums = [2,1,3,4] **Output:** false **Explanation:** There is no sorted array once rotated that can make nums.

Example 3:

Input: nums = [1,2,3] **Output:** true **Explanation:** [1,2,3] is the original sorted array. You can rotate the array by x = 0 positions (i.e. no rotation) to make nums.

Constraints:

* `1 <= nums.length <= 100` * `1 <= nums[i] <= 100`

Code Snippets

C++:

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

}
};
```

Java:

```
class Solution {
public boolean check(int[] nums) {

}
}
```

Python3:

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