Question : 1752. Check if Array Is Sorted and Rotated

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 A[i] == B[(i+x) % A.length], where % is the modulo operation.

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 = 3 positions to begin on the 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.

Link: https://leetcode.com/problems/check-if-array-is-sorted-and-rotated/description/

Explanation

Array

$[1, 2, 3, 4, 5]$

$(1 < 2 < 3 < 4 < 5)$

(And)

$(5 > 1)$

An array is sorted when the Previous element is smaller than its forwarded element

And
when the last element is greater thant from its first element

Now,

[See observations]

original : $[1 \overset{\le}{,} 2 \overset{\le}{,} 3 \overset{\le}{,} 4 \overset{\le}{,} 5 \overset{>}{]}$

Rotate to right by i++; where as i=0;

① $[5 \overset{\textcircled{>}}{,} 1 \overset{\le}{,} 2 \overset{\le}{,} 3 \overset{\le}{,} 4 \overset{\le}{]}$

② $[4 \overset{\le}{,} 5 \overset{\textcircled{>}}{,} 1 \overset{\le}{,} 2 \overset{\le}{,} 3 \overset{\le}{]}$

③ $[3 \overset{\le}{,} 4 \overset{\le}{,} 5 \overset{\textcircled{>}}{,} 1 \overset{\le}{,} 2 \overset{\le}{]}$

④ $[2 \overset{\le}{,} 3 \overset{\le}{,} 4 \overset{\le}{,} 5 \overset{\textcircled{>}}{,} 1 \overset{\le}{]}$

⑤ $[\underbrace{1 \overset{\le}{,} 2 \overset{\le}{,} 3 \overset{\le}{,} 4 \overset{\le}{,} 5 \overset{\textcircled{>}}{]}}_{\text{Original one}}$

Invalid

$$\Rightarrow \overset{\times}{\overbrace{[\;2 \overset{>}{,} 1 \overset{<}{,} 4 \overset{>}{,} 3 \overset{<}{,} 5\;]}}$$

$$\Rightarrow [\;4 , 2 , 1 , 5 , 3\;]$$

✡ These are not Produced by rotation of an array by n position.

Conclusion:

~~In a rotated~~

* In a rotated array, there will always be only one pair of elements where the previous element is greater

arr [n-1] >arr[n] -(only one)

Code:
To determine the array was originally sorted and then - rotated.

⇒ Run a loop from (i=1; i<n; i++)

⇒ check for the condition

```
if (arr [n-1] > [n]) {

        count++;
  }
```

Special case

=> Check condition for the last and first element. This is necesary because the loop does not compare them.

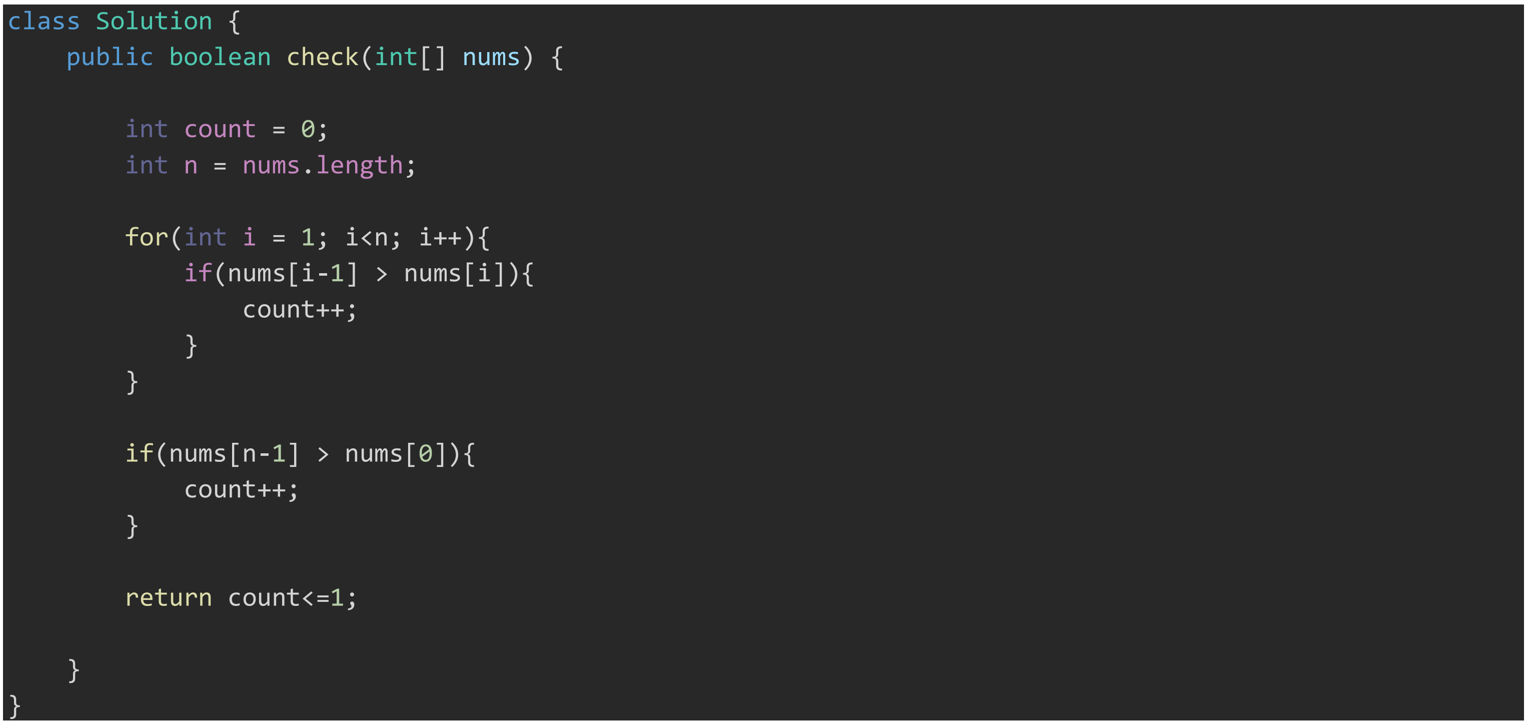
outside the loop.

```
if (arr[n] > arr[0])
        Count++;
    }
```

=> if array contains all identical elements like [1, 1, 1, 1], the Count will never increment So, this will not work at all
[return Count ==1;]

That is why we use this

```
return Count <=1;
```



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

        int count = 0;
        int n = nums.length;

        for(int i = 1; i<n; i++){
            if(nums[i-1] > nums[i]){
                count++;
            }
        }

        if(nums[n-1] > nums[0]){
            count++;
        }

        return count<=1;

    }
}
```