



# Explanation

## Walkthrough on the examples

### Example 1: `nums = [3,4,5,1,2]`

Iterate circular pairs:

- compare 2→3: `2 > 3?` No.
- compare 3→4: `3 > 4?` No.
- compare 4→5: `4 > 5?` No.
- compare 5→1: `5 > 1?` Yes → `count = 1`.
- compare 1→2: `1 > 2?` No.

End: `count = 1` → return `True`. (This matches rotation of `[1,2,3,4,5]`.)

### Example 2: `nums = [2,1,3,4]`

Pairs:

- 4→2: `4 > 2?` Yes → `count = 1`.
- 2→1: `2 > 1?` Yes → `count = 2` → function returns `False` immediately.

Two decreases → cannot be one rotation of a sorted array.

### Example 3: `nums = [1,2,3]`

Pairs:

- 3→1: `3 > 1?` No.
  - 1→2: `1 > 2?` No.
  - 2→3: `2 > 3?` No.
- `count = 0` → sorted (rotation by 0) → return `True`.

## The `%` (modulo)

The modulo (`% n`) is there **so that when `i = 0`, the “previous index” wraps around to the last element** — making the comparison circular.

So this ensures that when comparing each `nums[i]` to its previous value, even the **first element** gets compared with the **last one**, completing the rotation check.

Without the modulo, you'd miss that last connection — which is crucial for detecting whether the array is a rotated version of a sorted list.