21. Check if array is sorted and rotated

Medium Accuracy: 38.92% Submissions: 13658 Points: 4

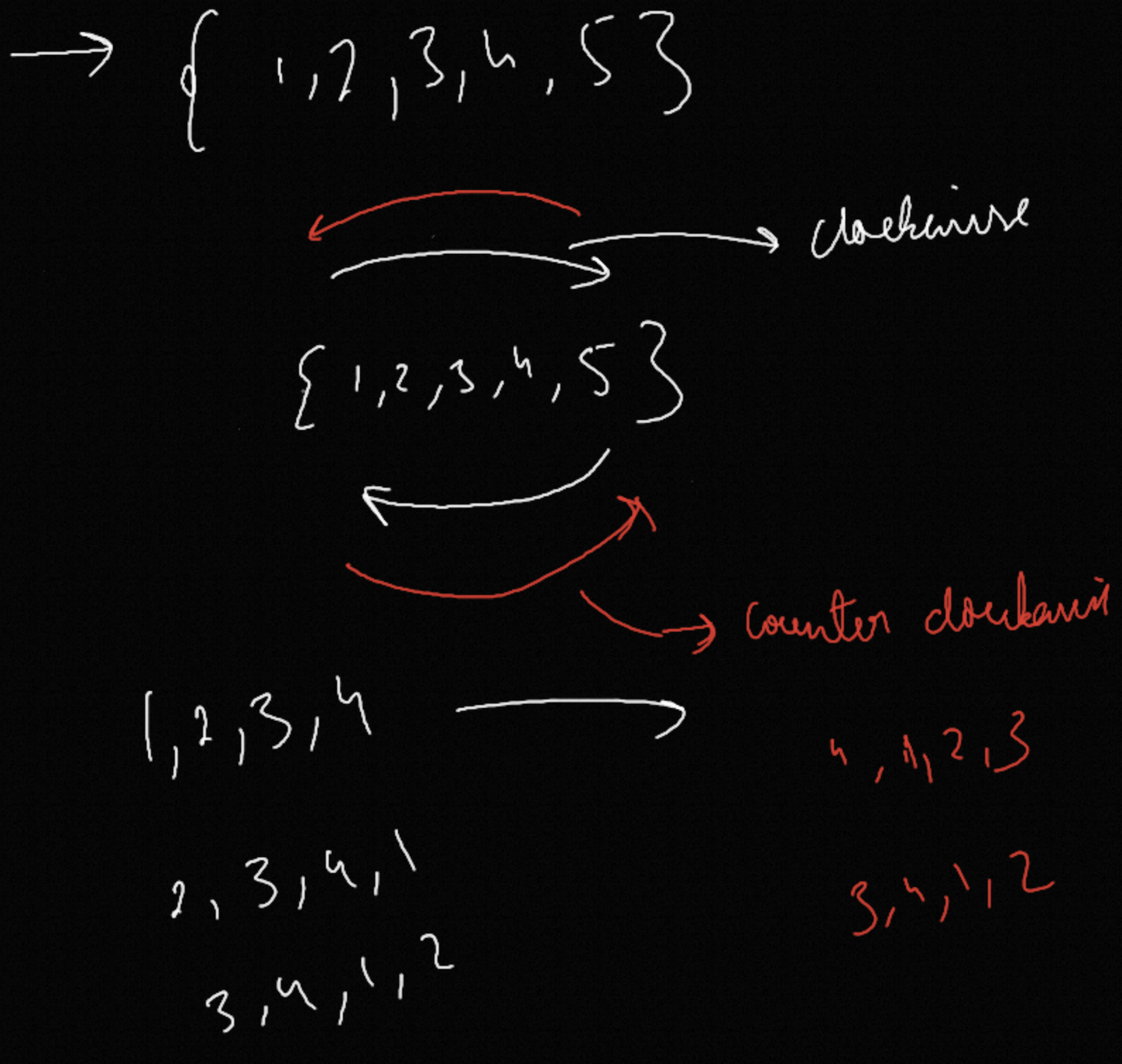
Given an array **arr[]** of **N** distinct integers, check if this array is **Sorted (non-Increasing or non-decreasing)** and **Rotated** counter-clockwise. Note that input array may be sorted in either increasing or decreasing order, then rotated.
A sorted array is not considered as sorted and rotated, i.e., there should be at least **one** rotation.

Example 1:

```
Input:
N = 4
arr[] = {3,4,1,2}
Output: Yes
Explanation: The array is sorted
(1, 2, 3, 4) and rotated twice
(3, 4, 1, 2).
```

Example 2:

```
Input:
N = 3
arr[] = {1,2,3}
Output: No
Explanation: The array is sorted
(1, 2, 3) is not rotated.
```



① Solution approach →

① check for increasing till you find first non decreasing one array count reached (i++)

↓

if $i \to n-1$ → false array only sorted not rotated

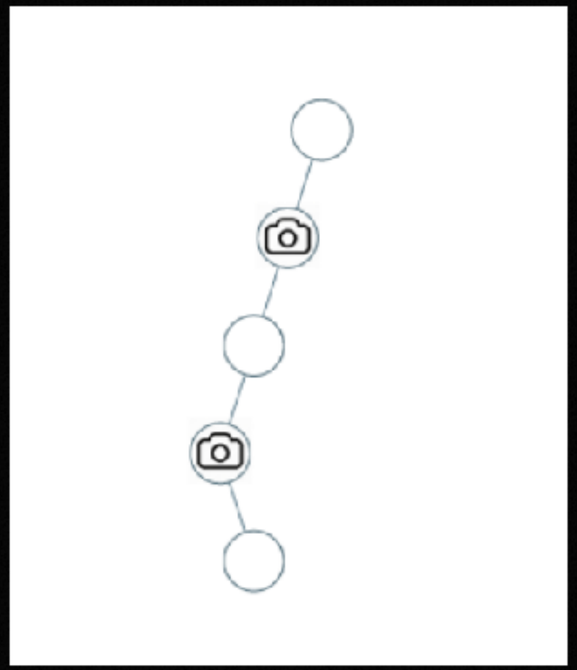
↓

$i < n-1$

↳ check for $\{i+1, n-1\}$ — if sorted

↓

None then Yes



```
//Function to check if array is sorted in increasing order and rotated.
bool II (int arr[], int n)
{
    int i = 0;
    //We use a loop to check whether elements are in increasing order and
    //stop at position where we find a smaller number than previous one.
    while (i < n - 1 and arr[i] <= arr[i + 1]) i++;

    //If we reach the end of the array, we return false.
    if (i == n - 1) return false;

    i++;
    //Now we check whether all remaining elements are in increasing order.
    while (i < n - 1 and arr[i] <= arr[i + 1]) i++;

    //If we reach the end and the last element is smaller than or equal to
    //first element we return true else we return false.
    if (i == n - 1 and arr[n - 1] <= arr[0])
        return true;
    else
        return false;
}
```