

## Check if reversing a sub array make the array sorted

Given an array of  $n$  distinct integers. The task is to check whether reversing **any one sub-array** can make the array sorted or not. If the array is already sorted or can be made sorted by reversing any one subarray, print **"Yes"**, else print **"No"**.

### Examples:

**Input :**  $arr [] = \{1, 2, 5, 4, 3\}$

**Output :** Yes

*By reversing the subarray  $\{5, 4, 3\}$ , the array will be sorted.*

**Input :**  $arr [] = \{1, 2, 4, 5, 3\}$

**Output :** No

### Method 1: Brute force ( $O(n^3)$ )

Consider every subarray and check if reversing the subarray makes the whole array sorted. If yes, return True. If reversing any of the subarrays doesn't make the array sorted, then return False. Considering every subarray will take  $O(n^2)$ , and for each subarray, checking whether the whole array will get sorted after reversing the subarray in consideration will take  $O(n)$ . Thus overall complexity would be  $O(n^3)$ .

### Method 2: Sorting ( $O(n \cdot \log(n))$ )

The idea is to compare the given array with its sorted version. Make a copy of the given array and sort it. Now, find the first index and last index in the given array which does not match with the sorted array. If no such indices are found (given array was already sorted), return True. Else check if the elements between the found indices are in decreasing order, if Yes then return True else return False