Strivers-A2Z-DSA-Sheet-main\02.Binary Search\1D Arrays\05.Check_If_array_is_sorted.cpp

```cpp
/*
QUESTION:
Given an array arr[] of size N, check if it is sorted in non-decreasing order or not.

APPROACH:
- We can use a recursive approach to check if the array is sorted in non-decreasing order or not.
- The base case for recursion is when the subarray has only one element or when the subarray is empty, in which case we consider it to be sorted.
- For a non-empty subarray, we compare the middle element with its next element. If they are in non-decreasing order and both the left and right subarrays are also sorted, then we consider the entire array to be sorted.
- We recursively check the left and right subarrays using the same approach.
- If any of the recursive calls returns false, we return false. Otherwise, we return true.

Example:

Input:
N = 5
arr[] = {10, 20, 30, 40, 50}
Output: 1
Explanation: The given array is sorted.

CODE:
*/

bool solve(int arr[], int low, int high) {
    if (low >= high)
        return true;

    int mid = low + (high - low) / 2;
    if (arr[mid] <= arr[mid + 1] && solve(arr, low, mid) && solve(arr, mid + 1, high))
        return true;

    return false;
}

bool arraySortedOrNot(int arr[], int n) {
    return solve(arr, 0, n - 1);
}

// TIME COMPLEXITY: O(log N)
// SPACE COMPLEXITY: O(log N) (for recursion stack)

```