1. 134. Given an array of integers nums, sort the array in ascending order and return it. You must solve the problem without using any built-in functions in O(nlog(n)) time complexity and with the smallest space complexity possible.

PROGRAM:-

```
def merge_sort(nums):
    # Base case: If the array is empty or has one element, it is already sorted
    if len(nums) <= 1:
        return nums

    # Divide the array into two halves
    mid = len(nums) // 2
    left_half = nums[:mid]
    right_half = nums[mid:]

    # Recursively sort each half
    left_sorted = merge_sort(left_half)
    right_sorted = merge_sort(right_half)

    # Merge the sorted halves
    return merge(left_sorted, right_sorted)

def merge(left, right):
    merged = []
    i = j = 0

    # Merge the two sorted lists into one sorted list
    while i < len(left) and j < len(right):
```

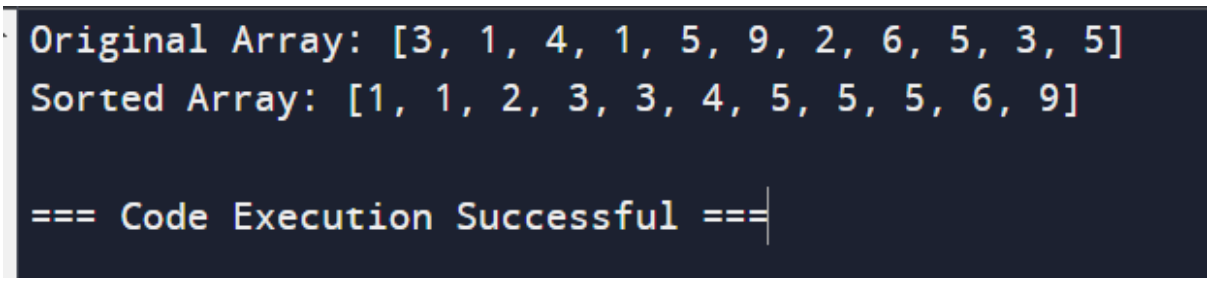
```
        if left[i] <= right[j]:
            merged.append(left[i])
            i += 1
        else:
            merged.append(right[j])
            j += 1

    # Append any remaining elements
    while i < len(left):
        merged.append(left[i])
        i += 1
    while j < len(right):
        merged.append(right[j])
        j += 1

    return merged

# Test Case
nums = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]
sorted_nums = merge_sort(nums)
print(f"Original Array: {nums}")
print(f"Sorted Array: {sorted_nums}")
```

OUTPUT:-



TIME COMPLEXITY:-O(n log n)