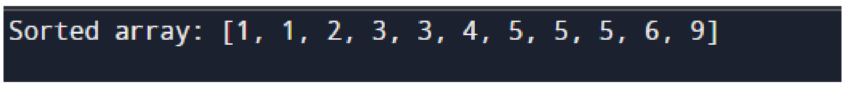
136.Given an array of integers nums, sort the array in ascending order and return it. You must solve the problem without using any built-in functions in O(nlog(n)) time complexity and with the smallest space complexity possible.

AIM: To sort an array in ascending oreder by given time complexity:

PROGRAM:

```
def merge_sort(nums):
   if len(nums) <= 1:
      return nums

   mid = len(nums) // 2
   left_half = merge_sort(nums[:mid])
   right_half = merge_sort(nums[mid:])

   sorted_nums = merge(left_half, right_half)

   return sorted_nums

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

   while i < len(left) and j < len(right):
      if left[i] <= right[j]:
         result.append(left[i])
         i += 1
      else:
         result.append(right[j])
         j += 1
      result.extend(left[i:])
   result.extend(right[j:])

   return result
nums = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]
sorted_nums = merge_sort(nums)
print("Sorted array:", sorted_nums)
```

OUTPUT: 

```
Sorted array: [1, 1, 2, 3, 3, 4, 5, 5, 5, 6, 9]
```

TIME COMPLEXITY: O(n log n)