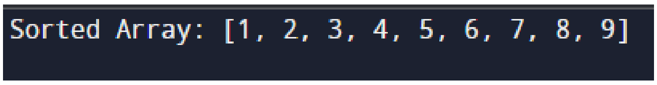
18. Given an array of integers nums, sort the array in ascending order and return it.You must solve the problem without using any built-in functions in O(nlog(n)) time complexity and with the smallest space complexity possible.

PROGRAM:

```
def merge_sort(nums):
   if len(nums) <= 1:
      return nums
   mid = len(nums) // 2
   left_half = nums[:mid]
   right_half = nums[mid:]
   left_half = merge_sort(left_half)
   right_half = merge_sort(right_half)
   return merge(left_half, right_half)


def merge(left_half, right_half):
   merged = []
   i = j = 0
   while i < len(left_half) and j < len(right_half):
      if left_half[i] < right_half[j]:
         merged.append(left_half[i])
         i += 1
      else:
         merged.append(right_half[j])
         j += 1
   while i < len(left_half):
      merged.append(left_half[i])
      i += 1
   while j < len(right_half):
      merged.append(right_half[j])
      j += 1


   return merged
```

```
nums = [5, 2, 9, 3, 7, 1, 8, 4, 6]

sorted_nums = merge_sort(nums)

print("Sorted Array:", sorted_nums)
```

OUTPUT: 

```
Sorted Array: [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

TIME COMPLEXITY: O(n log n)