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.

```
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)
         sorted_nums = merge(left_half, right_half)
    return sorted_nums
def merge(left, right):
    merged = []
    i = j = 0
    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
        merged.extend(left[i:])
    merged.extend(right[j:])
    return merged
```

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

sorted_nums = merge_sort(nums)

print(sorted_nums)

output:

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

=== Code Execution Successful ===
```

TIME COMPLEXITY:-O(n log n)