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.

Code:

```
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_half, right_half):
    merged = []
    left_index, right_index = 0, 0
    while left_index < len(left_half) and right_index < len(right_half):
        if left_half[left_index] < right_half[right_index]:
            merged.append(left_half[left_index])
            left_index += 1
        else:
            merged.append(right_half[right_index])
            right_index += 1

    while left_index < len(left_half):
        merged.append(left_half[left_index])
        left_index += 1
    while right_index < len(right_half):
        merged.append(right_half[right_index])
        right_index += 1
```

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

Output:

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

Time Complexity:

- T(n)= O(n)