80. Merge sort

Code:

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

def merge(left, right):
    sorted_arr = []
    i = j = 0
    while i < len(left) and j < len(right):
        if left[i] < right[j]:
            sorted_arr.append(left[i])
            i += 1
        else:
            sorted_arr.append(right[j])
            j += 1
    while i < len(left):
        sorted_arr.append(left[i])
        i += 1
    while j < len(right):
        sorted_arr.append(right[j])
        j += 1
    return sorted_arr
arr = [38, 27, 43, 3, 9, 82, 10]
sorted_arr = merge_sort(arr)
print(f"Sorted array: {sorted_arr}")
```

Output:

```
Sorted array: [3, 9, 10, 27, 38, 43, 82]
```

Time Complexity:

- T(n)= O(nlogn)