Assignment-03
Explaination

Task-1

The code is using merge sort algorithm to sort the list in ascending order. Here time complexity is O(nlogn) in every case(best, average, worst).

Task-2

The code finds the maximum value from the given list using a recursive function similar to merge sort algorithm. Here time complexity is O(nlogn) as it is using merge sort algorithm.

Task-3

The code counts the number of inversions from the list using enhanced merge sort algorithm instead of brute force algorithm for more efficient time complexity. Here the time complexity is O(nlogn) which is more efficient than $O(n^2)$.

Task-4

The code finds the maximum value of the sum of an integer and the square of the maximum integer in its right subarray for each integer in the list using merge sort algorithm. Here time complexity is $O(n \log n)$.

Task-5

The code is using quick sort algorithm to sort the given list in ascending order. Here the time complexity is $O(n \log n)$ for best and average case. But in worst case $O(n^2)$

Task-6

The code finds the k-th smallest element from the list ~~using~~ without sorting using the partition function of quick sort