

10:22/4/026

## Task 1

In this code, I used merge sort algorithm to sort the list. It splits into smaller segments, sort those and merges them. The time complexity is  $O(n \log n)$ .

## Task 2

In this task, I used divide and conquer method by ~~recursively~~ recursively divide into smaller segment and find the maximum value using the divide and conquer. The time complexity also  $O(n \log n)$ .

### Task 3

I used merge sort algorithm to count the number of inversions in the list. The inversion count is the number of pairs where the element on the left is greater than the element on the right.

### Task 4

Here, I applied a modified merge sort algorithm and calculated the maximum value by adding sequence of elements from the right subarray to the max value from the left subarray. ( $AR[i] + AL[i]$ ).



## Task 5

Here, I used Quick Sort algorithm to sort the list in ascending order. In Quick Sort, the  $k$ th min or maximum value is found and inserted in the  $k$ th index by swapping.

## Task 6

Here, I used Quick Sort on the list to find the  $k$ th smallest element for each query. It recursively narrows down the search to find the  $k$ th smallest val. Time complexity is  $O(n)$ .