

Ishraq, Kamal Adib

ID: 22299325

Section: 20

### Task 1

The merge sort function divides the array into two half and sorts them recursively and we are using two pointers to sort.

### Task 2

Here in this function we are using merge sort and we are storing the max values in a variable and comparing them each other to find the max.

### Task 3

using the merge function of merge sort algorithm whenever a value from right array enters the sorted we count how many element in the left array is still remaining, then we add the number in the count.

## Task 4

using the merge function and merge algorithm before we merge the two arrays we take the greatest element of left array and from the right array we also take <sup>same</sup> value of values and compare the 3 subarrays one from left, one from right and from mid.

## Task 5

in quick sort we <sup>take</sup> the leftmost element as pivot and we choose lower bound and upper bound in the partition function using two pointers to place the pivot on the right index



## Task 6

Here in the quick <sup>select algorithm</sup> ~~sort~~ we take the rightmost element as the pivot, then check if the position of pivot is the position we want and then return it. Until we call the recursive methods.