

Explanation

Lab-03

* Task-01:

In the Merge Sort algorithm, there are two functions - mergeSort and merge. MergeSort is called to divide the array recursively and for length of 1 array, it returns the array. By calculating the middle index of the array, it divides the array into left and right parts recursively. Then merge function is called on them ~~what~~ to check which part is larger and sort the elements accordingly. Any left out elements are appended to the end of the sorted array and return the sorted array.

Task-02:

Here I write a modified function of mergeSort named find-max. Before calling the function I initialize $\text{left} = 0$ and $\text{right} = \text{array size} - 1$. Then I call the find-max function which recursively find out the maximum element of given array using the algorithm of merge sort.

Task-03:

Here I use divide and conquer approach to solve the problem. The countPair function use merge sort algorithm which checks the given conditions of subarrays using ~~counts~~ ^{CheckAndMerge} function. ~~At~~ ~~the~~ ~~counts~~ ^{function} uses merge algorithm which returns the merged array and total count of subarrays which fulfil the given conditions to the countPair function.

Task-04:

Here I use same approach I used in task-3. The findMax function use merge sort algorithm which checks the given conditions through CheckAnd Merge function which returns max value to the find-Max function.

Task-05:

Here I use ^{given} pseudocode to solve the problem. Before calling the function quicksort I took 0 as low parameter and $N-1$ as high parameter.

Task-06:

Here I write find-Kth-Smallest function which uses the partition function of task-5 to find the index number of Kth smallest number of given list. Then it returns the number.

