

Name: Abu Tardous Shawon. ID: 2201020

Task-1: First I take the length and the array from input and then convert all the value into integer. Then I call Mergesort function to find the single element. Then I use merge function to add the value in the new list by checking the ascending order.

Then I returned the new sorted list.

Task 2: From the input I convert the array into integer values. Then I call ~~max~~ ~~max~~ max num function. It will follow divide and conquer rule. And if check every single element with other elements. And finally returns the max value.

Task 3: After converting all values of array into integer. I take a variable count to count the pairs. Then I run a nested loop and check if it fulfill the condition. Then I increase the value of count. Then I ~~return~~ print the count.

Task 4: I called mergesort to find ~~size~~ divide the list. After that it will pair two array to merge function. After adding two ~~array~~ ^{array}, I run a loop to calculate the sum of two value. And if compare it with max. And when length of the list is

sorted equal to $b_{1:2}$ then I returned the max value.

Task 5: I called quick sort to sort the array.

In this function I called Partition. It will arrange the array with respect to ~~the~~ pivot. Then It will returned ~~in~~ the index of pivot. And by ~~the~~ ^{array} recursively call quick sort will sort the ~~function~~ ^{array} ~~function~~.

Task 6: I made two array from the input file. One is ~~of~~ the main array and another array contains quintiles. Then I pass two array in ~~find-kth-smallest~~ ~~find-kth-smallest~~ function. In this function ~~for~~ it will append the smallest value in a list by calling ~~km~~ ^{km} smallest function. In this function It call partition to find the pivot. It will returned ~~at~~ the index of the index of pivot and ~~km~~ ^{km} smallest function ~~else~~ will find smallest value by recursively calling the function.