

Task 1

- # Generating the list (newList)
- # apply merge sort.
- # Then Merge function will return a sorted list after merging "a" & "b" list.
- # Creating output.

Task 2

- # Generating the list.
- # dividing the list until we get only 1 element list
- # Then we conquer the two list.
- # And in the conquer part, method, I return a list which have highest value
- # finally creating output

Task 3

Generating the newList.

use only one loop, which condition is "pointer1 < len(newList) - 2".

then checking that the 1st pointer value is greater than to the 2nd pointer value or not. If it is true, then I incrementing the value of "count" variable.

Also iam checking that pointer2 is in the last index or not.

If pointer2 is in the last index, then I increment the value of pointer1. & again pointer2 will be the next position of pointer1 ($\text{pointer2} = \text{pointer1} + 1$).

~~# And every~~

finally creating output file.

Task 4

logic is same like what I did in Task 3.
I just simply checking the given condition, and finding the maximum sum.

Task 5

- # use Quick sort
- # Calling the partition function.
 - moving the l towards right
 - moving the r " left.
 - if $l \leq r$; then swap.
 - then putting the pivot in its perfect position.
- recursive call left side
- " a Right "

Task 6

→ Going through each of the N -th position.

→ Removing all the element that are between $1-(N-1)$.

→ Finally, finding the N -th position value and append it in the list.

→ Output file creating.