

Task-1

I have written the mergesort algorithm to sort the list in $O(n \log n)$ time.

Task-2

I have used the divide and conquer part of the ~~merge~~ mergesort algorithm to return the max value at every ~~rec~~ recursive step.

Task-3

here I have slightly changed the ^{value} merge function in the mergesort algorithm. In the condition where an element of the left array is bigger than the right array element, I have added a count that adds as many elements there are on the right side of designated left sub-array.

Task - 4

I have used two small functions that returns the max value at every recursive step.

(same as task-2). One of them squares the value. Then the answer function recursively adds the smaller element and the square of the bigger element. In each recursive step it evaluates and returns the largest value.

Task - 5

I have used the quicksort algorithm as shown in the pseudo code.

Task - 6

Here I have used the quick select algorithm. It makes changes in the mother function. It does recursion on the left or on the right based on the returned value being bigger or smaller than the pivot.