Gebze Technical University Computer Engineering

# CSE222/Homework 6 Report

Student name:Yağız Hakkı Aydın
No:1901042612

## Part one : Definition of the problem and requirements

### a)Definition of the problem:

**Three algorithms are required to be implemented.**

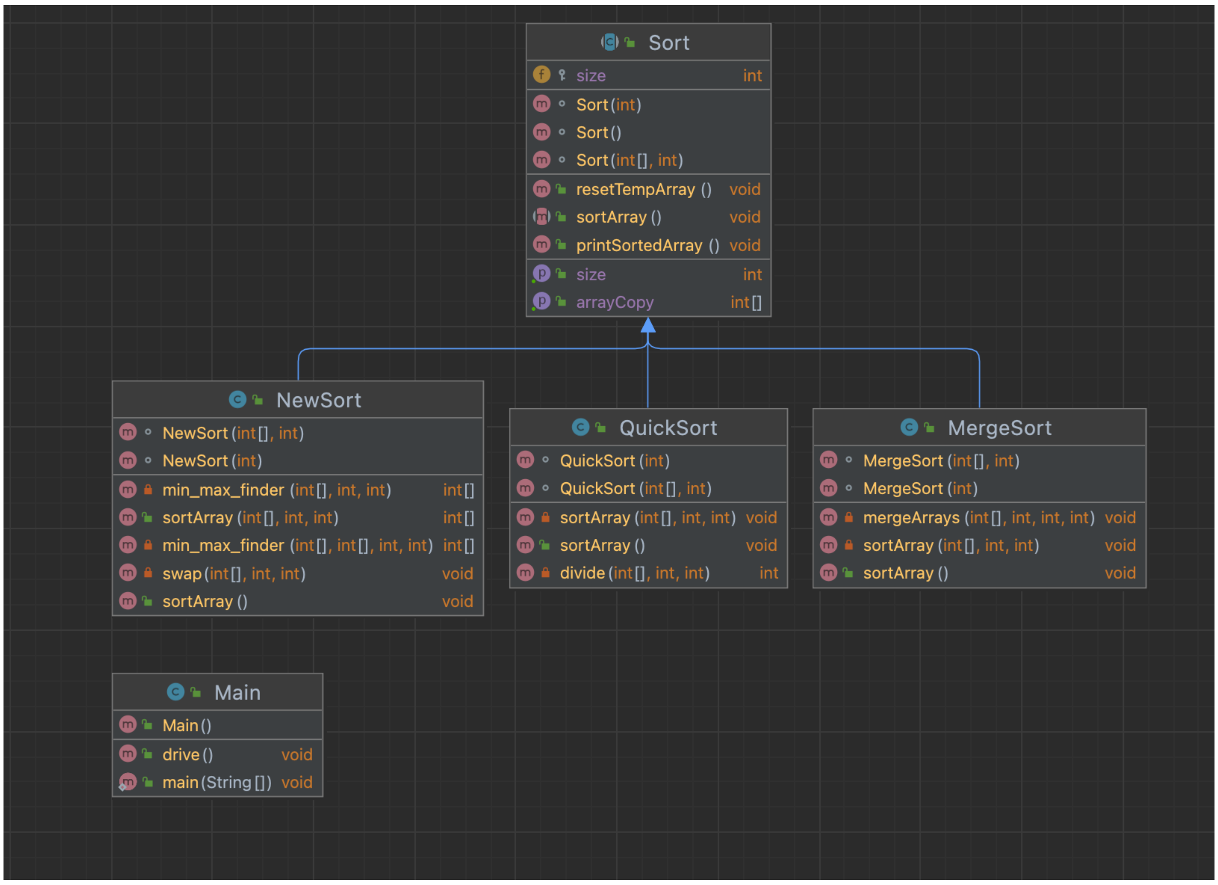
### a)Requirements:

->Implementing merge sort algorithm

-> Implementing quick sort algorithm

-> Implementing new sort algorithm which is given as pseudo code

## Part two : Class Diagrams



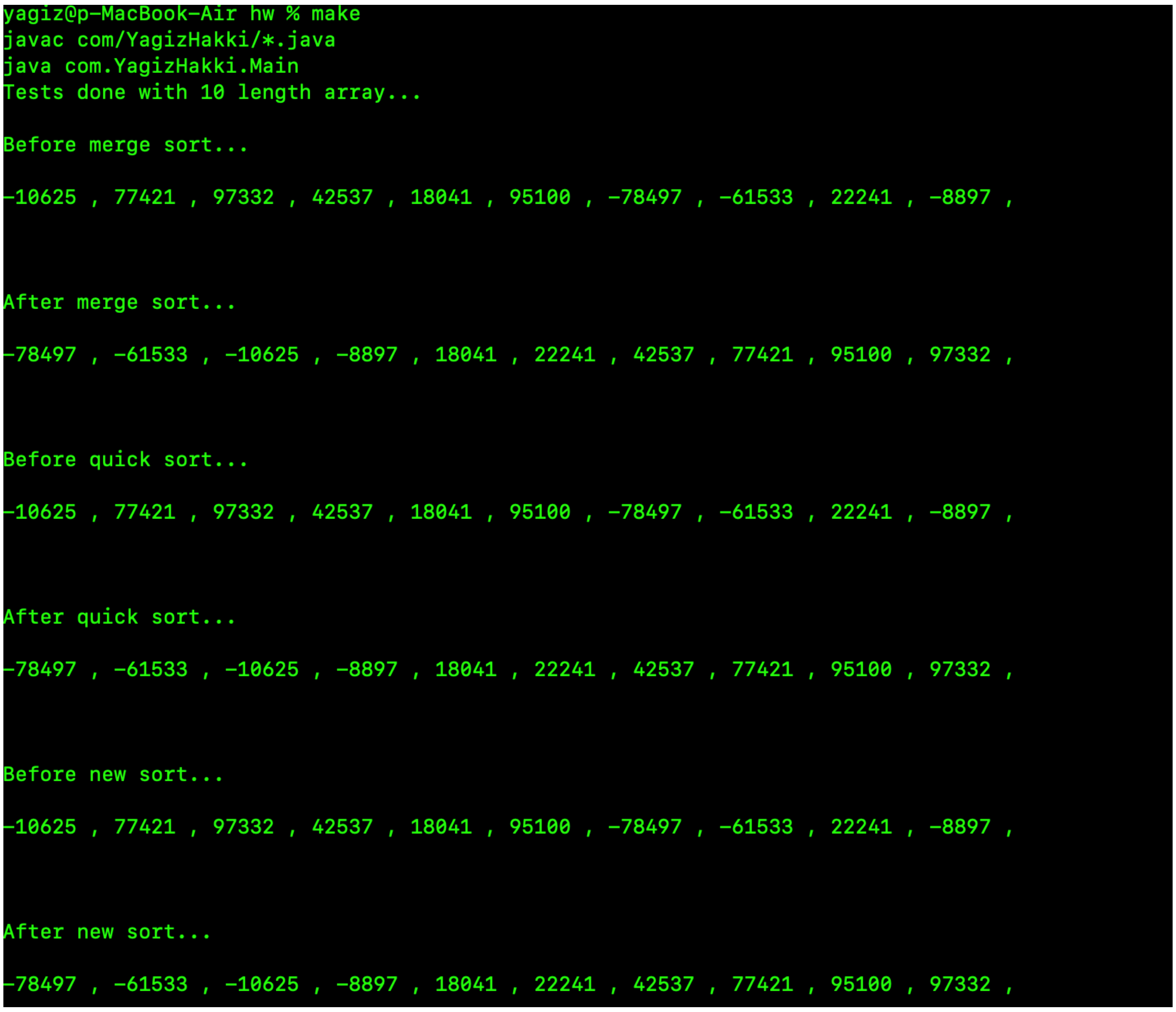
## Part three : Problem solution approach

Newsort algorithm is same with given code.For finding min and max, min_max_finder() function runs recursively and returns an array with size 2.First element is minimum element’s index and second element is for maximum element.

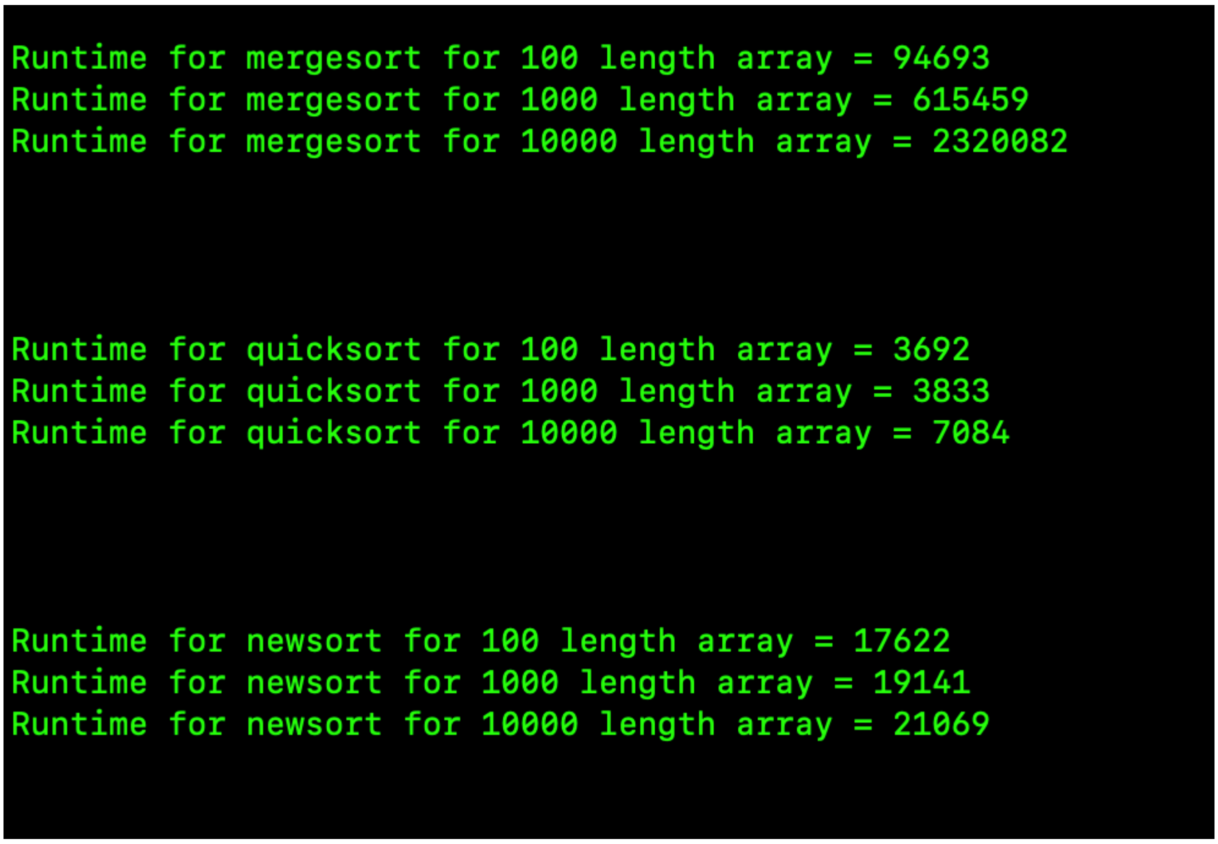
For merge sort, sortArray() function is called for seperate two pieces of given array recursively until dividing array as much as possible and these pieces gets merged again.

For quick sort,a pivot point is choosen at every step and left and right side of this point gets sorted.

## Part four : Test Cases



```
yagiz@p-MacBook-Air hw % make
javac com/YagizHakki/*.java
java com.YagizHakki.Main
Tests done with 10 length array...

Before merge sort...

-10625 , 77421 , 97332 , 42537 , 18041 , 95100 , -78497 , -61533 , 22241 , -8897 ,



After merge sort...

-78497 , -61533 , -10625 , -8897 , 18041 , 22241 , 42537 , 77421 , 95100 , 97332 ,



Before quick sort...

-10625 , 77421 , 97332 , 42537 , 18041 , 95100 , -78497 , -61533 , 22241 , -8897 ,



After quick sort...

-78497 , -61533 , -10625 , -8897 , 18041 , 22241 , 42537 , 77421 , 95100 , 97332 ,



Before new sort...

-10625 , 77421 , 97332 , 42537 , 18041 , 95100 , -78497 , -61533 , 22241 , -8897 ,



After new sort...

-78497 , -61533 , -10625 , -8897 , 18041 , 22241 , 42537 , 77421 , 95100 , 97332 ,
```



```
Runtime for mergesort for 100 length array = 94693
Runtime for mergesort for 1000 length array = 615459
Runtime for mergesort for 10000 length array = 2320082



Runtime for quicksort for 100 length array = 3692
Runtime for quicksort for 1000 length array = 3833
Runtime for quicksort for 10000 length array = 7084



Runtime for newsort for 100 length array = 17622
Runtime for newsort for 1000 length array = 19141
Runtime for newsort for 10000 length array = 21069
```

Quick sort and merge sort must have same complexity(  O(n*logn)  )

Tests are done with 10 size array to Show that sorting algorithms work.But times are measured with asked array lenghts.