05 LAB - QuickSelect, QuickSort, MergeSort

You must implement the following algorithms QuickSelect, QuickSort, and MergeSort.

You must have three java files: Quick.java, MergeSort.java, Driver.java

Quick.java

```
This file should have the following methods:

// Returns the Kth smallest element in the array
public static int QuickSelect(int[] data, int start, int end,
int k){

}

// Sorts the array
public static void QuickSort(int[] data, int start, int end){

}

// Creates the partitions
public static int partition (int[] array, int start, int end){

}
```

Note: You may implement a swap method to avoid repeated code.

Optional: Dutch National Flag Algorithm to create partitions.

MergeSort.java

```
// Sorts the array
public static int[] MergeSort(int[] data){

}

// Merge two sorted arrays. You may use any of these two
options:
public static int[] merge(int [] left, int[] right){

}

public static void merge(int[] destination, int [] left, int[]
right)
```

Optional: Another merge method using only two arrays: `int [] left and int[] right.`

Driver.java

- Call the sorting methods from your Driver.java (QuickSelect, QuickSort and MergeSort).

- Randomly generate the array that will be sent as a parameter when you call the methods.

- Create at least three test cases.

Specifications

- Do not use an ArrayList
- Sorting algorithms should be recursive