CME 2204 Assignment-1
(Comparison of Merge sort and Quick sort)

This Homework Is Due By 11:55 PM on Friday **April 07, 2023**

Rules:

- The submissions will be checked for the code similarity. Plagiarism will be graded as zero.
- You are required to upload two different files. One is the source code you have written in Java programming language and the report.
- The files you are required to upload are given below with explanations:

  (STUDENT_NUMBER)_(STUDENT_NAME)_HW1_Code
  (Source code you have written for the HW1)

  Example = 2043901815_Augusta_Ada_HW1_Code

  (STUDENT_NUMBER)_(STUDENT_NAME)_HW1_Report.pdf

  Example = 2043901815_Augusta_Ada_HW1_Report.pdf

1. In this assignment you will be implementing and testing all two sorting algorithms: Merge sort and Quick Sort. You will write a class and own methods. Sorting algorithm methods will be invoked from this class;

```
public class SortingClass {

        mergeSort(int[] arrayToSort, String numberOfPartitions) { .... }
        and quickSort(int[] arrayToSort, String pivotType) { .... } .
}
```

In the ***mergeSort method***, you must implement two different ways of choosing number of partitions (***numberOfPartitions***).

- **TwoParts** : recursively split the array into 2 parts.
- **ThreeParts** : recursively split the array into 3 parts.

In the ***quickSort method***, you must implement three different ways of choosing the pivot (***pivotType***) as given below.

- **FirstElement :** The pivot is always the first element of the array to be sorted.

- **RandomElement :** The pivot is chosen at random from any element in the array to be sorted.
- **MidOfFirstMidLastElement :** The pivot is chosen to be the element whose value is in the middle among the {first, middle, and Last} elements in the array to be sorted.

2. You are expected to compare each sorting algorithm according to the running times (in milliseconds) for different inputs and fill the following table accordingly.

	EQUAL INTEGERS			RANDOM INTEGERS			INCREASING INTEGERS			DECREASING INTEGERS		
	1,000	10,000	100,000	1,000	10,000	100,000	1,000	10,000	100,000	1,000	10,000	100,000
mergeSort **TwoParts**												
mergeSort **ThreeParts**												
quickSort **FirstElement**												
quickSort **RandomElement**												
quickSort **MidOfFirstMidLastElement**												

- You must create the input arrays by yourself in the Test class according to the specified rules (1000, 10000, 100000 of each equal, random, increasing, and decreasing order).
- Note that, the elapsed time of creating the arrays should not be considered during the calculation of the total running time of sorting algorithms. You only have to check how long the sorting algorithm is running.
- One of the methods you can use to measure the time elapsed in Java is given below:

```
n=1000     // array size
int arrayToSort = new int[n];
for(int i = 0 ; i < n ; i++)

        arrayToSort[i] = i;   // generate numbers in increasing order

long startTime = System. currentTimeMillis();
mergeSort(arrayToSort,"TwoParts");// run one of the sorting methods
long estimatedTime = System. currentTimeMillis() - startTime
```

3. Prepare a scientific report in the light of the results obtained from your program and the information you learned in the class. Establish a connection between the asymptotic running time complexity (in Θ notation) and the results (in milliseconds) of your practices. Your report should include the Comparison table. Also, you are expected to determine the appropriate sorting algorithms for the following scenarios. For each of the following scenarios, which algorithm would you choose and why? Explain your thoughts and reasons clearly and broadly.

a. You have a Turkish-English dictionary with a single word for each word in alphabetical order and you want to translate it into an English-Turkish dictionary. For example; if your Tur-Eng dictionary contains [ayı : bear, bardak : glass, elma : apple, kitap : book] then your Eng-Tur dictionary should be [apple : elma, bear : ayı, book : kitap, glass : bardak]. If we think that there are thousands of words in your dictionary, which sorting algorithm do you use to do this translation faster?

b. When you inquire Sub-Upper Pedigree, an ordered list of people according to their birth date comes out in the system of e-Devlet. However, you want to rank the people from the youngest to the oldest one. If you are asked to do this operation using a sorting algorithm, which algorithm do you use?

Grading Policy

Item	%
MergeSort	20
QuickSort	25
RunningTime	20
3.Part (a and b)	15
Report	20