Alexandria University
Faculty of Engineering
Computer and Systems Engineering
Department

File Structures Course
Assigned: Wednesday, March 6, 2024
Due: Saturday, March 16, 2024

Assignment 1
Implementing Sorting Techniques

1 Introduction

The goal of this assignment is to become familiar with the different sorting techniques.

2 Sorting Techniques

- You are required to implement 3 different sorting techniques. You're also required to compare the running time performance of these algorithms against each other:
  - An O(n^2) sorting algorithm such as Selection Sort, Bubble Sort, or Insertion sort.
  - An O(n lg n) sorting algorithm such as Merge Sort or Quick sort algorithm in the average case.
  - An O(n) sorting algorithm such as Counting Sort or Radix sort algorithm.

  Select one of the sorting algorithms from each class mentioned above.

3 Requirements

3.1 Sorting Algorithms

You are required to implement a Sort_Array class that can encapsulate a list of elements and sort them using the different algorithms. You need to implement the following methods in this class:

1. Initialize (constructor): Takes an input file structured containing the list of elements comma separated, reads it, and initializes the array.

2. Simple Sort: Takes 1 parameter which is whether to return the intermediate arrays or return the final sorted array only. Then applies the sorting algorithm you chose to implement from the O(n^2) sorting algorithms and either return the intermediate results or the final result.

3. Efficient Sort: Takes 1 parameter which is whether to return the intermediate arrays or return the final sorted array only. Then applies the sorting algorithm you chose to implement from the O(n lg n) sorting algorithms and either return the intermediate results or the final result.

4. Non-Comparison Sort: Takes 1 parameter which is whether to return the intermediate arrays or return the final sorted array only. Then applies the sorting algorithm you chose to implement from the O(n) sorting algorithms and either return the intermediate results or the final result.

3.2 Command Line Interface

You should implement a command line interface that will enable us to deal with the implemented sorting algorithms. This interface must take the path of the file containing the list of elements as an initial input and then create the sorting object using it. The interface will be composed of a main menu that allow the user to apply subsequent operations on the array from the following list:

1. Sorts the list of elements of the array. You should allow the user to choose 1 of the 3 algorithms to run. The user can ask for the sorted array or/and the intermediate arrays as well.

3.3 Java Unit Testing

You should provide a set of 15-20 JUnit tests that test the correctness and efficiency of the different implemented parts. Also, these tests must show a comprehensive comparison between the 3 algorithms w.r.t time factor. For each algorithm, try examples where the sorting technique will perform in the worst case, best case, and average case as well.

4 Notes

- You need to work in teams of 4 or 5.
- You need to use Java in your implementation.
- Each team should submit via teams the code and a **report** explaining the time and space analysis of the 3 algorithms in addition to a comparison between them at different array sizes w.r.t the meantime to get the sort of the entire array.