School of Computer Sciences

CPT 212 Design & Analysis of Algorithms

# ASSIGNMENT I : Principles of Analysis of Algorithms and Sorting Methods Semester II 2024/2025

## Objectives

- Manipulate data structures or algorithms in problem solving and programming.
- Perform complexity analysis of algorithms.

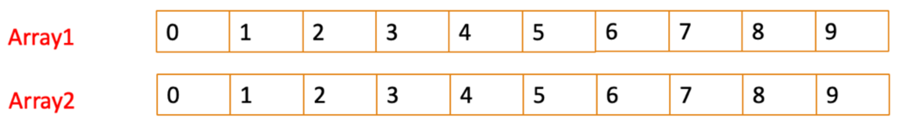
## Specification

The figure below shows how a sorting algorithm sorts a list of number.

```
int[ ] x = {275, 087, 426, 061, 409, 170, 677, 503}
for (int i = 0; i < x.length; i++){
```

1. Initialize

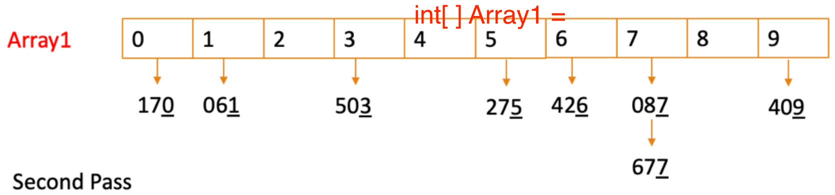
Example: 275, 087, 426, 061, 409, 170, 677, 503



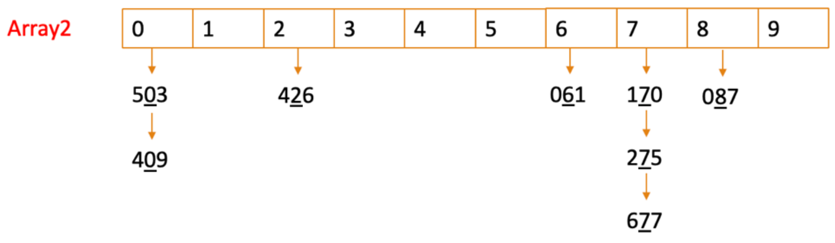
2. Iteration

First Pass

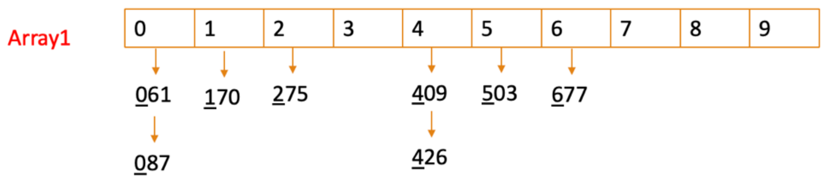
int[ ] Array1 =



Second Pass



Third Pass



3. Reorder

Sorted list: 061 087 170 275 409 426 503 677

UNIVERSITI SAINS MALAYSIA

**School of Computer Sciences**

**CPT 212 Design & Analysis of Algorithms**

1. Study and understand the algorithm in the figure, and write a Java program that sort **exactly** as the figure described. Any deviation from the step, no mark will be given. Example, the figure shows arrays in Initialization, and you are not using arrays, then no mark is given for the initialization step. Add comments in the source codes according to the explanation of the figure. Assumption: positive integer with the same length. (40%: Initialization=5%, Iteration, first pass=10%, second pass=10%, subsequent pass=10%, Reorder=5%)

2. Modify the sorting algorithm above to sort a list of word. A word is a string that makes up of alphabets [a-z]. A word consists of at least one character, and the length may not be the same. The Java program must be a modification of the earlier program, and not a different implementation. A different implementation will get zero mark. (30%)

3. Perform complexity analysis on both the algorithms by experiments. Count the number of primitive operations performed by the algorithm by adding counters to the program. Plot the graph: number of primitive operations vs n. Determine the big-O for both algorithms based on the results of the experiment. (30%)

**This assignment is to be carried out in a group of 3 (maximum). References taken from any sources must be quoted and declared. No sharing of answers with other groups. Penalty for late submission, no excuse for late submission will be accepted.**

- **Program and report submission deadline**: **Sunday, 11/5/2023 11.59 pm.**
- **Only one of the group leader has to submit the work.**
- **Report to me if there is any group member that didn't do their work.**

UNIVERSITI SAINS MALAYSIA

**School of Computer Sciences**

**CPT 212 Design & Analysis of Algorithms**

## Assignment Assessment Rubric

| | Excellent (80-100%) | Good (65-79%) | Moderate (40-64%) | Poor (0-39%) | Total |
|---|---|---|---|---|---|
| Part 1: Sorting algorithms (40%) | Algorithm is implemented according to the steps given. Codes are clearly commented. | Algorithm is implemented according to the steps given. Codes are not commented. | - | Algorithm does not implement according to the steps given. | |
| Part 2: Modification (30%) | Algorithm can sort words. It is the modification of the first algorithm. Algorithm is clearly commented. | Algorithm can sort words. It is the modification of the first algorithm. Algorithm is not clearly commented. | Algorithm can sort words in some cases and fail in some cases. It is the modification of the first algorithm. Algorithm is clearly commented. | Algorithm is not a modification of the earlier algorithm or program cannot run. | |
| Part 3: Analysis (30%) | All counters are correctly added. Graphs are plotted. Correctly specify the time complexity of the algorithms and correctly justify by analyzing the graphs obtained. | Some counters are incorrectly added/ not added. Graphs are plotted. Correctly specify the time complexity of the algorithms, but some of the graphs obtained in the experiments are not analyzed. | Many counters are incorrectly added/ not added. Some graphs are incorrectly plotted. Correctly specify the time complexity of the algorithms but do not analyze the graphs obtained. | No counter is added. No graphs. No time complexity given. No analysis being carried out. | |