School of Computer Sciences

CPT 212 Design & Analysis of Algorithms

ASSIGNMENT I : Principles of Analysis of Algorithms and Sorting Methods
Semester II 2022/2023

Objectives

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

Specification

1. Convert the radix sort algorithm that is described in the Appendix section to Java program. Note: You must implement exactly like what is described. (40%)
2. Modify the radix sort algorithm to sort floating point number. The program must be a modification, and not a different implementation or variant of radix sort. (30%)
3. Perform complexity analysis on both the algorithms by experiments. Plot a graph: number of operation vs n. Determine the big-O for both algorithms. (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, 14/5/2023 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

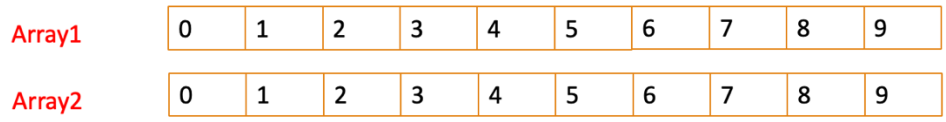
CPT 212 Design & Analysis of Algorithms

Assignment Assessment Rubric

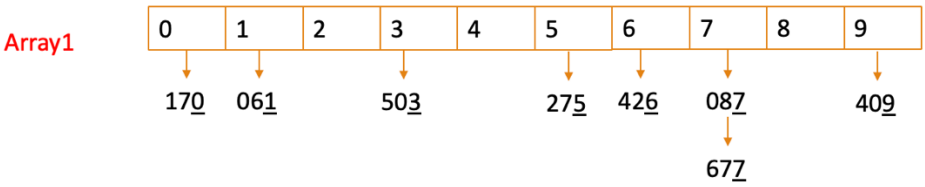
	Excellent (80-100%)	Good (65-79%)	Moderate (40-64%)	Poor (0-39%)	Total
Part 1: Sorting algorithms (40%)	Algorithm is implemented according to the steps given. Codes are clearly commented. Good OO programming practices are applied.	Some steps in the algorithm are not implemented correctly. Codes are not clearly commented. OO programming practices are applied.	Many steps in the algorithm are not implemented correctly. Codes are not clearly commented. No OO programming practices are applied.	Algorithm does not implement according to the steps given.	
Part 2: Radix sort (30%)	Algorithm can sort floating point numbers. It is the modification of the first algorithm. Algorithm is clearly commented.	Algorithm can sort floating point numbers. It is the modification of the first algorithm. Algorithm is not clearly commented.	Algorithm can sort 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 Algorithm 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.	

Radix Sort: Example

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

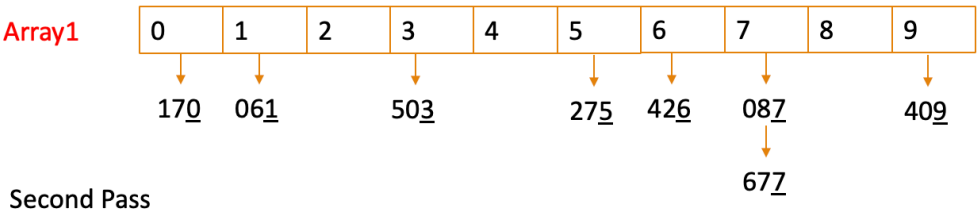


First Pass

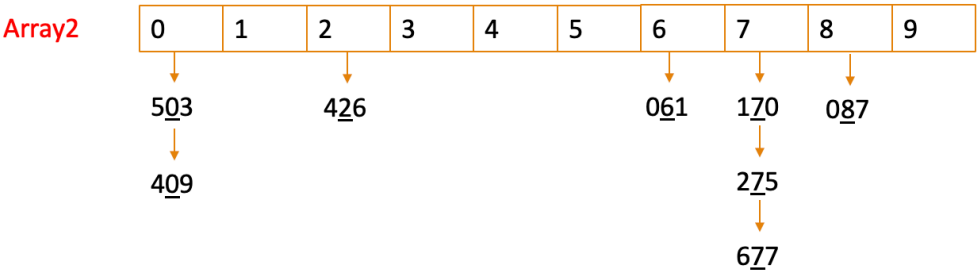


Radix Sort: Example

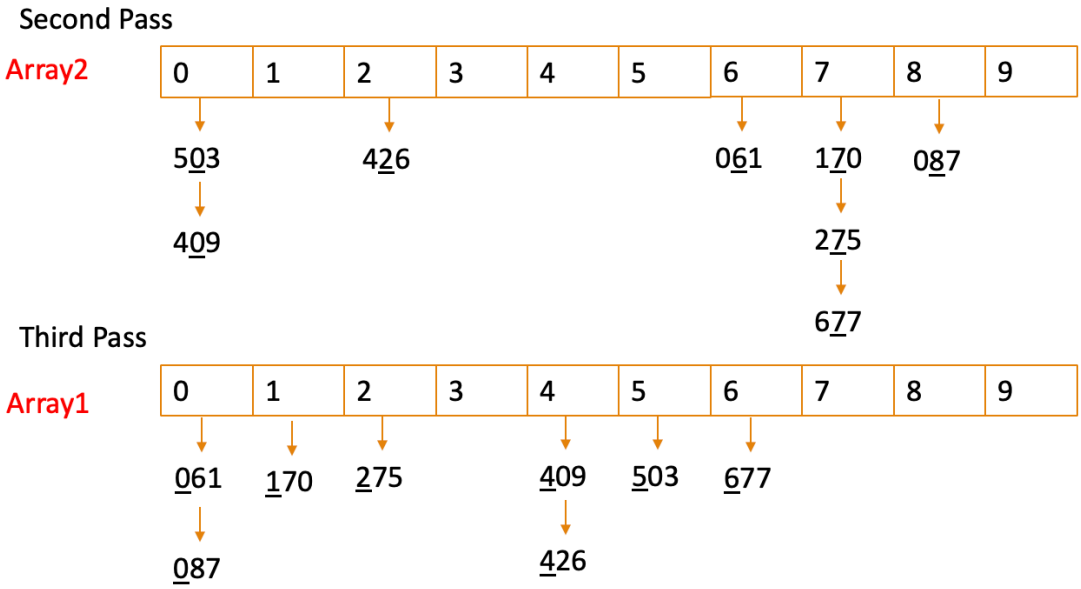
First Pass



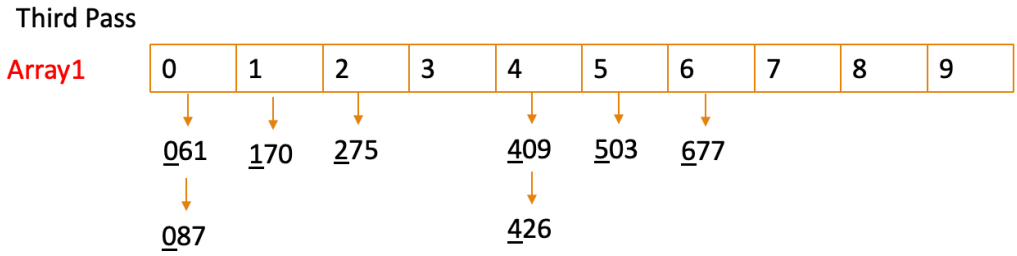
Second Pass



Radix Sort: Example



Radix Sort: Example



Sorted list: 061 087 170 275 409 426 503 677