

## Tutorial 04: Arrays Sorting Algorithms and Analysis

### Related Objectives from Unit Outline:

- Describe (arrays, linked lists, binary trees, and hash tables) data structures and analyze the complexity and performance of their associated algorithms

### Objectives:

1. To demonstrate the awareness of the principles of algorithms in array merging.
- 

### Tasks:

Complete the following.

**Task 1:** Consider the problem of reading a file of (unsorted) values into an array, where

the array must be sorted. There are  $n$  values in the file.

- (1) Write an algorithm to read all of the unsorted values into the array, and then sort the array using the selection sort algorithm given in Algorithm 3.27. What is the time efficiency of your algorithm?
- (2) Write an algorithm to read each value in turn, and insert it into a sorted array (initially empty). What is the time efficiency of your algorithm? How does this compare with your answer to part (1)?

**Task 2:** You are given two unsorted arrays of values. You are required to obtain a sorted array containing all these values. Suggest two different ways of achieving this. Compare their time efficiency. (Note: Assume that suitable merging and sorting algorithms are already available.)

**Task 3: Test the Java selection and insertion sorting programs given in WS0401**

(Download the Java code from Blackboard)

- a. Run this program to observe the sorting process using pre-coded data;
- b. Explain the executed results according to the principles of the selection and insertion sorting algorithms;
- c. Modified the code to test different **data sets**.

**Task 4: Test Java merge and quick sort programs given in WS0402**

(Download the Java code from Blackboard).

- a. Run this program to observe the sorting process using pre-coded data;

- b. Explain the executed results according to the principles of the merge and quick sort algorithms;
- c. Modified the code to test different ***data sets***.

**Task 5: Test the Vector class using WS0403** (Download the Java code from Blackboard).

- a. Note how to construct vector objects;
- b. Observe the operation of some vector methods by analyzing the executed results.