Tutorial 4

Introduction

In this tutorial, students are asked to implement the sorting algorithms including Selection sort, Insertion sort, Bubble sort, Merge sort. Students may refer to the example section to see the implementation code of 4 mentioned sorting algorithms. Class SortComparison implements all sorting algorithm together and demonstrates the running time of each algorithm.

In the exercise sections, students will try to use sorting algorithm to solve several related problems.

Examples

1. **Example 01 – Selection sort.**

Student should create a class that implements the Selection sort algorithm, with the following functions:

- Ask user to input n, array A using the keyboard
- Sort the array A using Selection sort algorithm
- Show the result

2. **Example 02 – Insertion sort**

Student should create a class that implements the Insertion sort algorithm, with the following functions:

- Ask user to input n, array A using the keyboard
- Sort the array A using Insertion sort algorithm
- Show the result

3. **Example 03 – Bubble sort**

Student should create a class that implements the Bubble sort algorithm, with the following functions:

- Ask user to input n, array A using the keyboard
- Sort the array A using Bubble sort algorithm
- Show the result

4. **Example 04 – Merge sort**

Student should create a class that implements the Merge sort algorithm, with the following functions:

- Ask user to input n, array A using the keyboard
- Sort the array A using Merge sort algorithm

- Show the result

5. Example 05 – Sorting algorithm comparison

Student should create a class that implements all the mentioned sorting algorithms, with the following functions:

- Randomly generated an array of the size N, copy the generated array to a new array
- Sort the new array using 4 sorting algorithms. Every time, before doing sorting, you need to redo the copying from the source array to make sure that you don’t sort an already sorted array.
- Measure and show the running time of each sorting algorithm.

Exercises

1. Exercise 1

Given an array A that stores N integers including some negative integers, some positive integers and one zero element. We want to rearrange the array so that, every negative element goes to the left, following is the zero element and then all the positive elements. Please write a Java program to solve this problem. Your program should:

- Ask user to input n, array A using the keyboard.
- Re arrange the array A.
- Show the result.

2. Exercise 2

Given a list of N students including student’s name and student’s mark (an integer in the range 0..10). We want to find M students who has the highest marks. Please write a Java program to solve this problem. Your program should:

- Ask user to input n, list of student including name and mark.
- Ask user to input an integer m
- Show the name of m students who has the highest mark.

3. Exercise 3

Please try to design an algorithm with the time complexity O(n) to solve the problem in Exercise 1.