Lab 3 – CS380 Algorithms and Data Structures

Sorting Algorithms Part 1

Download the SortIntegers.java file. It contains the implementation of Bubble Sort, Insertion Sort, and Selection Sort.

Carefully review the code, understand how each algorithm works, and run them.

TASK 1

1. Run all three algorithms for an array of size **n = 1000**.
2. Count how many times a **swap** is performed in **Bubble Sort** and **Selection Sort**. *(Note: Insertion Sort does not perform swaps.)*
3. Compare the results. Is there a difference? Explain why this occurs.

TASK 2

Choose **one of the three sorting algorithms** and modify it so that it can **sort an array of strings** instead of integers.

TASK 3

You are given an array of customers. Each customer is represented by the following Java class:

```java
class Customer {
    String name;
    String surname;
    int age;

    Customer(String name, String surname, int age) {
        this.name = name;
        this.surname = surname;
        this.age = age;
    }

    void display() {
        System.out.println(name + " " + surname + " (" + age + ")");
    }
}
```

And in the main function we create an array of 10 customers like this:

```
Customer[] customers = {
    new Customer("John", "Smith", 30),
    new Customer("Alice", "Brown", 25),
    new Customer("Bob", "Adams", 28),
    new Customer("Diana", "Clark", 35),
    new Customer("Diana", "Davis", 22),
    new Customer("Adam", "Smith", 51),
    new Customer("Mary", "Walsh", 18),
    new Customer("Adam", "Reed", 32),
    new Customer("Carl", "English", 29),
    new Customer("Bob", "Gates", 20)
};
```

Using **one of the three sorting algorithms** of your choice, modify it to implement a function that can:

1. Sort all customers by **first name (ascending)**.
2. Sort all customers by **age (ascending)**.
3. Sort all customers **first by first name, and then by surname** if the first names are the same.

Quick Reminder:

Customers is an array of Customer objects. For example:

customers[2] refers to the **third customer** in the list.

customers[2].name returns the name of the third customer (e.g., "Bob").