COMP610 — Data Structures and Algorithms

Lab 03

What to do

Do all the tasks, and answer all the questions. For code, make sure you follow the Code Laws. You can work individually or in pairs. Record non-code answers, as you may have to give open-class feedback on them.

Before you start

Review the lecture material on sorting. Be sure that you understand the *principles* behind bubble sort, selection sort and merge sort. Do them on paper to make *sure* you understand the way they work.

In this lab, you will be working with *static classes*. These classes exist only to provide one or more static methods — thus, they have no fields, and their constructor is private, takes no arguments, and does nothing. The idea is that these classes never get instantiated; instead, we simply use their methods. You've already used a few static classes in the past (think `System.out`).

Question 1

Download the Eclipse project for this lab, unzip it, and open it in Eclipse. Inside, you should find a `BubbleSorter` class, several tests, and a `Profiler` class.

1. Rename the packages to fit the Code Laws.
2. Run `BubbleSorterTest`; ensure all tests pass.
3. Using `BubbleSorter` as a guide, implement a `SelectionSorter` class. It should be a static class, implementing a `sort` method, which copies the input array, then sorts it using *selection sort*. Ensure all the tests in `SelectionSorterTest` pass.
4. Implement a `MergeSorter` class. It should be a static class, implementing a `sort` method, which copies the input array, then sorts it using *merge sort*. Ensure all the tests in `MergeSorterTest` pass. Be careful — this is harder than it looks!

Question 2

1. What is the time complexity of each of the sorts that you've written? What in the code suggests this?
2. Which of these sorts do you think will run fastest? Slowest? Why?
3. Run `Profiler` and check the table it outputs. Does this support your answers above? If not, how does it differ?