EECS 2030: Lab 4

(1.5 % of the final grade, may be done in groups of up to three students)

Motivation

The purpose of this lab is to complete two implementations of *selection sort* and compare their performance to that of a modern algorithm.

Part 1: Getting Started

Download a zip file containing the Lab 4 Eclipse project.

Import the project into Eclipse by doing the following:

1. Under the **File** menu choose **Import...**
2. Under **General** choose **Existing Projects into Workspace** and press **Next**
3. Click the **Select archive file** radio button, and click the **Browse...** button.
4. In the file browser that appears, navigate to your download directory (exactly where this is depends on what computer you working on; on the lab computers the file will probably appear in your home directory)
5. Select the file **lab4.zip** and click **OK**
6. Click **Finish**.

Explore the existing methods and the test cases, try to understand the purpose of each line. For example, how the generic types were used and how the `Comparator` types were specified.

Part 2: Sorting Implementation

Open the class `Lists` and implement the two methods with TODO sections.

For the recursive solution, you may use the implementation that was briefly described in the notes on recursion.

For the iterative solution, your implementation will basically contain two nested loops: for each location *i* of the result, you search for an appropriate element (i.e., the smallest one) in the locations from *i* to the end of the list.

JUnit testers for your classes are available in the project that you downloaded. Note that the testers are not very thorough, and it may not catch all errors that you might make. Also note that passing all of the tests in this tester *does not* guarantee a good solution (in other words, you should think critically about your implementation for each method).

Part 3: Experiments

Complete the implementation of the `sortExperiment` class. Use the existing pattern to add the other sorting algorithms into the experimental procedure, and format the output in a similar fashion. E.g.,

```
N=1000
Default Sort, Time spent: 1.76309 ms
Selection-Iterative, Time spent: XXX ms
Selection-Recursive, Time spent: YYY ms

N=10000
Default Sort, Time spent: 15.358378 ms
Selection-Iterative, Time spent: XXX ms
Selection-Recursive, Time spent: YYY ms

(repeat for larger values of N),
NOTE: you will probably discover that large values cause the
recursive solution to fail, and make the selection sort
algorithms take a very long time. Reduce the maximum value
of N in the sortExperiment accordingly.
```

Your program should print the results above to the console using appropriate method calls in your main method, or in additional static helper methods. Save the result of your program execution in a file `testrun.txt` and submit it together with your other files. The `testrun.txt` file can be produced by simply manually copying the output of your program from the console window into a text editor and saving it, or by using *output redirection*.

If you have questions, don't hesitate to post your questions on the course forum on Moodle, or contact the instructor directly (andriyp@cse.yorku.ca).

Grading

The assignment will be graded using *the Common Grading Scheme for Undergraduate Faculties*[1]. We look at whether the code passes the unit tests, satisfies the requirements of this documents, and whether it conforms to the code style rules.

Submission

Find all the `java` files in your project (there should be four of them: three classes and the test run) and submit them electronically via Moodle (no zipping is required).

If working in a group, make only one submission and include a **group.txt** file containing the names and the student numbers of the group members. The deadline is firm. Contact the instructor *in advance* if you cannot meet the deadline explaining your circumstances.

Academic Honesty

Direct collaboration (e.g., sharing your work results across groups) is not allowed (plagiarism detection software may be employed). However, you're allowed to discuss the assignment

[1] https://secretariat-policies.info.yorku.ca/policies/common-grading-scheme-for-undergraduate-faculties/

requirements, approaches you take, etc. Also, make sure to state any sources you use (online sources – including web sites, old solutions, books, etc.). Although using outside sources is allowed – with proper citing, if the amount of non-original work is excessive, your grade may be reduced.