CS 150: Project I

1 Project Description

The objective of this assignment is to study how the theoretical analysis of a variety of sorting algorithms compares with their actual performance. The sorting algorithms you will study are:

1. Selection sort
2. Bubble sort
3. Insertion sort
4. Merge sort
5. Three versions of Quicksort based on the choice of the pivot element

You will implement the above algorithms and compare their performances on carefully designed test cases with the results of theoretical analysis.

2 Assignments

The project is to be completed by a team of two (and if necessary, three).

2.1 Programming Portion

In the programming portion of the assignment, you will implement the above algorithms.

For Quicksort, you will implement three different strategies for selecting the pivot:

1. Pivot Choice 1: The first element in the array.
2. Pivot Choice 2: A random element in the array.
3. Pivot Choice 3: The median of the first, middle, and last elements in the array.

Your program should follow the principles of the object-oriented design. Specifically, each sorting algorithm should be implemented as a generic class that implements a common interface `Sorter.java`. The experiments should be conducted in a separate class `ExperimentController.java`. The common components of Quicksort should be implemented in an abstract class `QuickSorter.java`, and the three versions of Quicksort will be implemented as concrete classes that extends the abstract class `QuickSorter.java`.

For the actual implementation of the above algorithms, you are allowed (in fact, expected) to borrow code from the textbook or the lecture notes with proper citations. But you are not allowed to borrow code from anywhere else, unless you have a compelling reason and obtain prior permission from me. You will need to make a few modifications to run the test cases and compare the running time. Most of your time should be spent on designing good test cases and analyzing your results in order to draw conclusions regarding the performance of the various algorithms.

2.2 Experimentation and Analysis Portion of Assignment

You should design your own test data carefully to expose the advantages and disadvantages of different sorting methods. The choice of test data is important. For example, you should consider the following questions: which input sizes should be tested, how many different inputs of the same size should be testes, which patterns of a given size should be tested (such as random, almost sorted, reversed order). Be smart about which experiments to run, i.e., don't run larger or more tests than you need to answer the questions reasonably well. Also, note that you will need to run your experiments several times in order to get stable measurements (times will vary depending on system load, input, etc.) The measurement of the running time should only include the actual sorting and exclude other tasks such as data generation and printing. Your write-up must include a coherent discussion of the design of your test data, the experimental results obtained from them, and the analysis of the results.

The general guideline for wiring the project report is on the course Moodle. The following are some guidelines specific to this project:

1. Introduction - This introduces the purpose of the project. What are the goals and how will you achieve these goals? For each of the sorting algorithms you are comparing, give a brief introduction, conduct a theoretical analysis of its running time (in best, worst, and average cases), and discuss its advantages and disadvantages.

2. Approach - This section describes your approach to solving the problem. How did you design the test cases you would use? How did you generate the data? Do you use the same inputs for all sorting algorithms? Why would these test cases expose the advantages and disadvantages of different sorting algorithms?

3. Methods - This section describes your experimental setup. What kind of machine did you use? What timing mechanism? How many times did you repeat each experiment? What times are reported?

4. Data and Analysis - This section describes the data that you obtained (plotted) and your analysis of the data. The questions you should address include: Which of the three pivot choices of Quicksort seems to perform the best in your tests? Graph the running time as a function of the input size for the three pivot choices. How did different algorithms perform on the test cases? Graph the running time of all algorithms as a function of the input size. How do your implementations of Merge sort and Quicksort compare to the implementations of the same algorithms in the Java standard library? Do your results agree with the theoretical analysis given in the introduction? Give reasonable explanations of discrepancies between the theoretical analysis and the experimental results.

5. Conclusion. Discuss notable observations you made in this project. What conclusion did you draw from this project and what have you learned from it.

6. References - Books, websites, lecture notes, APIs that you used for the experiment/project. These references must be cited within the body of the report. Your bibliography should follow an established style (APA, MLA, Chicago, etc).

3 Grading

The major emphasis of this project is to measure and analyze the performance of the algorithms. Therefore, grading on this assignment will put the greatest weight on the choice of test data and the quality and insightfulness of your analysis of your results.

Your project will be graded on the following criteria (assuming the program compiles and runs):

1. correctness of the program (your program should perform correctly and efficiently)

2. documentation (methods and classes) including Javadoc
3. unit testing (each public method and public class should be tested)
4. object-oriented design
5. quality of the experiments and analysis
6. quality of the project report