CMPSC 112
Introduction to Computer Science II
Fall 2016

Laboratory Assignment Five: Studying the Efficiency of Incremental String Building

Introduction

The current module of the course has stressed the importance and purpose of both empirical and analytical evaluations of algorithm performance. For this laboratory assignment, we will replicate one of the experiments reported in your textbook as we learn more about the practice of and limitations associated with experimental studies. You will conduct your own experiments, collect tables of data, draw well-supported conclusions, and explain why your results are evident. Next, you will take the first step towards seeing the connection between the experimental and analytical evaluation of algorithms. Finally, you will continue to practice the use of software engineering tools.

Review Your Textbook

To do well on this laboratory assignment, you should read the content in Section 4.1, paying particularly close attention to the results in Table 4.1 and Figure 4.1. Please also examine the slides that we have discussed during our recent class sessions. If you have questions about this reading assignment or the material that was presented in class, then please see the instructor. If done appropriately, you may post your question to the `#laboratory` channel of our Slack team.

Using and Enhancing the String Experiment Framework

To start this laboratory assignment, you should return to the `cs112F2016-share` Git repository and type the "`git pull`" command in the terminal window. Now, you should have a `lab5/` directory that you can explore further. Once again, please make sure that you can find the source code in this new directory and you understand why the directories in the assignment are structured the way that they are. Next, you should use GVim to study the source code in the `build.xml` file. As in the past assignments, when editing a Java program you can type "`ant compile`" in your terminal window and it will compile the Java class and save the bytecode in the correct subdirectories inside of the `bin/` directory. Please see the course instructor if you cannot get this to work.

After you have carefully reviewed the source code for `StringExperiment.java`, you should compile and run this program, using the `build.xml` rules that correctly sets the `CLASSPATH` and references the Java program with its "fully qualified name". Please run the `StringExperiment` multiple times and note the data tables that it produces. You should be aware of the fact that some input sizes for this program may take a very long time to run! If that is the case, then you may want to reconsider the program's current implementation and change it so that the experiment runs more efficiently. What trends do you see in this data set? How do your tables of data compare to the one that the authors present on page 152 of your textbook? Can you clearly explain why these data values are evident in your data set and the one produced by the textbook's authors?

After you have tried out the `StringExperiment`, you will notice that it does not run the method under study for multiple trials and then compute summary statistics (i.e., the arithmetic mean and

the standard deviation) of the timings. Therefore, you should improve the program by moving the statistics calculation code from a previous laboratory assignment into your source code repository for this assignment. Then, you should carefully integrate this new code so that it runs multiple trials and calculates the arithmetic mean and standard deviation for the numerous timings. Again, you may want to think about the feasibility of running one or both algorithms for multiple trials and very long strings because it may take an inordinate amount of time to run the experiment.

It is worth pointing out that your textbook contains several useful insights into the pattern that you should follow when making observations about time overhead. For instance, when describing the results from running the `StringExperiment`, page 153 notes that "[A]s the value of n is doubled, the running time of `repeat1` typically increases more than fourfold." What does this suggest about the likely worst-case time complexity of the `repeat1` method? Additionally, page 172 includes the following statement when describing the performance of `repeat2`: "the running times in that table ... demonstrate a trend of approximately doubling each time the problem size doubles." Again, what would this observation suggest about the likely worst-case time complexity of `repeat2`?

Finally, using what you know about the implementation of Strings in the Java language, you should explain why one of the algorithms has substantially better performance than the other. Ultimately, you should submit the commented code of and results from using your experimentation framework that systematically doubles the size of the input to the `repeat1` and `repeat2` methods.

Carefully Review the Honor Code

The Academic Honor Program that governs the entire academic program at Allegheny College is described in the Allegheny Academic Bulletin. The Honor Program applies to all work that is submitted for academic credit or to meet non-credit requirements for graduation at Allegheny. This includes all work assigned for this class (e.g., examinations, laboratory assignments, and the final project). All students who have enrolled in the College will work under the Honor Program.

Summary of the Required Deliverables

This assignment invites you to submit a signed and printed version of the following deliverables:

1. A sample output from running all doubling experiments in all of their relevant configurations.
2. A description of all of the features supported by your own doubling experiment tool.
3. Using JavaDoc comments, the documented version of the `StringExperiment` source code.
4. A comprehensive written report that fully explains the results of your experimental studies.
5. A reflective commentary on the challenges that you faced when implementing your program.
6. A reflective commentary on the challenges that you faced when conducting the experiments.
7. Suggestions for the types of laboratory assignments that you would like to later complete.

Again, along with turning in a printed version of these deliverables, you should ensure that everything is also available in the repository that is named according to the convention `cs112F2016-<your user name>`. Please remember that students in the class are responsible for completing and submitting their own version of this assignment. You should see the instructor if you have any questions.