Lab 3

Due 11:59pm, 28 February

Handout 4
CSCI 136: Spring, 2005
21 February

Complexity and How Fast is Java?

1 Short Answers

Complete the following problems from the book and bring the answers to lab.

4.1
4.5
4.9
4.10
4.11

2 Lab Program

The goal of this lab is to complement our discussion of complexity analysis in lecture with measuring how long it takes to perform some operations in a Java program.

Do the laboratory at the end of Chapter 4. You should read and think about the lab before Wednesday. Since you will write only a few small programs, you do not need to bring a design.

Programming Hint: When you run your programs, use `java -Xint`. The `-Xint` flag will turn off dynamic optimizations. This will give you more accurate results.

The book describes writing a formal lab report. For us, it is sufficient to submit the programs used for experiments 2–7, and a README file including the following information for each experiment:

- A description of what was measured.
- Timings data. As described in the text, be sure to record the data for perhaps 5–10 times executions. Identify the *lowest* measured time, which should be the most accurate.

Additional notes:

- In step 7, use 4–5 reasonable choices for vector sizes. Creating vectors with several hundred to several thousand elements would be a good place to start.
- At the end of your README file, include a paragraph that (1) compares storing `int`s vs. `String`s into arrays; (2) compares the relative speeds of storing data into variables, arrays, and vectors; and (3) for step 7, compares the expected complexity ("Big O") with the observed timings.
- Also include answers to Thought Questions 2 and 3.

Submit all six programs and your README using `turnin` as usual.

2.1 LaTeX and Gnuplot

This section decribes some tools you are welcome to use to write up this lab, if you wish to write it up more formally.

The LaTeX typsetting tool is used to generate all handouts for this course, was used to write our textbook, and is the most widely-used system for generating technical research papers. An example LaTeX document can be found on the handouts webpage in `latex-ex.tar`.

Gnuplot is a tool to make graphs (see `http://www.gnuplot.org`). If interested, try out some of the examples from the gnuplot tutorial at `http://www.duke.edu/~hpgavin/gnuplot.html`.

The file `start.gp` from the handouts page contains a script for gnuplot to get you started. Once you generate a postscript file, use the command `open file.ps` to open it in a viewer. You can print it to the printer using the `lpr file.cps` command.

To practice, plot functions that are $O(1)$, $O(n)$, $O(n^2)$. In addition, copy the file `unknown.dat` from the web page. Plot the data in that file. (What do you think it represents?)