Lab 14 - Sort Detective

David B. Levine
Computer Science Department
St. Bonaventure University
Copyright, 2001, 2003

"I had the spots removed for luck; fortunately I remember where the spots formerly was." - Big Jule, *Guys and Dolls*

Objective

The primary objective of this lab is for you to apply your theoretical knowledge of sorting algorithms to solve a problem of poor user interface design. More specifically, you will be given a program which is designed to measure comparisons, data movements, and execution time for the four sorting algorithms discussed in class and one additional sorting algorithm. Unfortunately, the designer of the program did not label the buttons properly. You must apply your understanding of the general properties of the algorithms (and in some cases of the code used to implement them) to determine the proper labeling of the buttons.

The secondary objective of this lab is for you to gain experience writing a concise, but complete analysis of a system.

Background

As you know from class, if you double the size of the data set that you give to a quadratic (O(N^2)) algorithm, it will do four times the work; by contrast, an O(NlogN) algorithm will do a bit more than twice as much; and, a linear (O(N)) algorithm will do only twice as much work. As you also know, the characteristics of the input data set can affect the expected performance of some but not all of our sorting algorithms. Before you begin the lab, you should review the expected performance of the algorithms on various data sets by completing the Pre-Lab Worksheet.

The sorting algorithms under study this week include (in alphabetical order): bubbleSort, insertionSort, mergeSort, quickSort, and selectionSort.

Instructions - Warning: read all of the instructions before beginning!

1. In this lab there is no need to open the Eclipse environment. Download *Detective.jar* to your desktop. This is an executable jar file. To run the program just double-click the icon for the program on your desktop. The program should open a window with assorted fields and buttons. If you see this, you have the program running properly.

2. Your task is to determine which sorting algorithm is associated with each button. Devise a plan which will enable you to match the particular algorithms to the button names. Hint: It may make sense to try to divide the sorts into initial groups and then to work on each group separately.

3. Execute your plan, taking careful notes as you go.
4. Describe the results of your experiment in a summary document. Begin with a summary of the matching and then show the rationalization process that justifies it. A sample from a similar (but much simpler) lab on searching can be seen here.

A Note on Writing

There is no coding in this lab. Thus, you should expect that a significant portion of the lab grade for this lab will be determined by the quality of the writing of the report. This includes the completeness of the report, the clarity (and grammar) of the writing, and general presentation. In the past, some students who matched all six sorts correctly have received poor grades due to sloppy writing. Don't be one of them!

Some of the sorts are difficult to distinguish. A carefully outlined experiment may compensate for an error in these cases if the writing makes it clear that your conclusions/guesses are substantiated by the data.

Finally, remember that your report needn't detail every experiment you ran. Rather, it should give sufficient information to justify your conclusions. It is possible to write a very short report that is completely correct if your experiments are well-chosen. After you learn the matching, you might consider whether there was a shorter way to arrive at your conclusion!

To Hand In

Turn in your Pre-Lab Worksheet and the final report from Step 5 along with a cover page.

Assignment Type (see Academic Practices and Policies Document):

Help Policy in Effect for This Assignment: Group Project with Limited Collaboration

In particular, you may discuss the assignment and concepts related to the assignment with the following persons, in addition to an instructor in this course: any member of your group; any St. Bonaventure Computer Science instructor; and any student enrolled in CS 132.

You may use the following materials produced by other students: materials produced by members of your group.

You may use the following materials produced by other students: NONE.