Sorting Algorithms(Assignment 6)

James Tran
Chapman University
CPSC 350-01
jatran@chapman.edu

***Abstract*—For this project, I used four different sorting algorithms to test their runtime and see how optimal each can be be. The algorithms used were Bubble Sort, Selection Sort, Insertion Sort, and Quick Sort.**

I. Time Differences

When running one at a time, I used the time STL in C++ to track when each sorting algorithm started and ended. Bubble Sort took the longest, followed by Selection Sort, followed by Insertion Sort, and finally followed by Quick Sort which was the fastest. The time differences were not as drastic as I had hypothesized. Bubble Sort took less than 20 seconds to sort over 36,000 numbers. I expected this algorithm to take nearly a minute. Quick Sort was nearly instant; however I was surprised that Insertion Sort was so close. I thought that Selection Sort would take the same amount of time as Insertion Sort but it took a few seconds longer. When checking the task manager, each algorithm started to use over 90% of my CPU every time the program ran.

II. Tradeoffs

Bubble Sort clearly has the advantage in ease of implementation. It is a simple algorithm to write and is easier to remember than any of the other algorithms. However, it should only be done when a relatively small data set needs to be sorted. Insertion Sort would not be optimal over Selection or Quick Sort unless some of the data was already sorted. Quick Sort would be the best algorithm to use for large data sets, but it is much more difficult to implement and requires more memory due to its recursive nature.

III. Programming Langauge

Using C++ did not change the results too drastically from using Java. C++ was indeed a little faster, but the results would not vary significantly over different languages.

IV. Shortcomings of Empirical Analyis

Empirical Analysis can yield some helpful data, but it is not the most efficient method of analyzing algorithms. It requires time to run as well as resources which cost money. In addition, it depends on too many variables such as platform and machine capability.

[1] M. Goodrich, R. Tamassia, and D. Mount, "Data Structures & Algorithms," John Wiley & Sons Inc., Second Edition, 2011.