Comparisons of Different Sorting Algorithms

Justin Bindi
Schmid College of
Science and Technology
Chapman University
Orange, California 92866
Email: bindi@chapman.edu

***Abstract*—The time differences were much more drastic than I had expected. I was expecting there to be some time difference but not much because they almost all O(n*n) [n squared]. Each algorithm has its own unique tradeoff that makes the choice of which one to use up to the user.**

I. Discussion

A. Time Differences

The average time for each sorting method:

1) *BubbleSort:* about 5000 microseconds
2) *SelectionSort:* about 1500 microseconds
3) *InsertionSort:* about 1000 microseconds
4) *QuickSort:* about 200 microseconds

The time differences were very different than my expectations. I was thinking that the numbers would all be within a few hundred microseconds of each other. However, there is a big difference and between each of them. I was very shocked when I saw that BubbbleSort took so much longer than the next longest one.

B. Algorithm Tradeoffs

BubbleSort is by far the easiest sorting algorithm to code but the drawback is that its performance is very lousy. SelectionSort has a lot less swaps than BubbleSort does so the performance time is much lower. Unfortunately, the performance is still not that good. InsertionSort is a bit more complicated to code but the performance time is a bit better than SelectionSort. Also, if the data is pre-sorted, then the code could run in linear run time as opposed to n squared. QuickSort is the fastest but is also the most complex. It requires partitioning but can run in O(nlogn).

C. Shortcomings of This Empirical Analysis

The shortcomings of this empirical analysis are that the actual data is not a constant and that it depends heavily on the computer that one is using. The processing time is different every run of the test and one has to take the average value of the data. Unless there were a sufficiently large number of trials, the data is not extremely useful. Also, the processing times will be different for every computer and very dependant on what is going on in the computer while the program is running.

II. Conclusion

The time differences were much more drastic than expected. BubbleSort is easiest to program and worst for performance. QuickSort was the fasted of algorithm but the hardest to program.

References

[1] Jain, Sandeep. A Computer Science Portal for Geeks. *GeeksforGeeks*, www.geeksforgeeks.org/.