Suleiman Karkoutli
Data Structures
Rene German
May 18th, 2018

Sorting Algorithms Analysis

Overall, this assignment has showed me the benefits and disadvantages of using empirical analysis to understand and evaluate the efficiency of algorithms. One benefit was that testing algorithms empirically did not require a lot of effort. I simply had to time each algorithm and look at how long it took to run. I did not need to evaluate the worst case scenarios for each algorithm using Big-Oh notation. Although empirical analysis is fairly easy to implement, it depends on the processor, software, and hardware of the particular computer that is being used to test the algorithms. This means that the results have the potential of being skewed and biased. What surprised me the most was that the sorting algorithms did not differ too much in terms of runtime. For instance, using a small set of data Bubble Sort ran at 5-6 milliseconds, QuickSort ran at 3-4 milliseconds, SelectionSort ran at 4-5 milliseconds, and InsertionSort ran at 4-5 seconds. The fastest algorithm implemented was Quicksort which makes sense since it has the fastest Big-Oh runtime of O(nlog(n)). Although Quicksort is the fastest both empirically and mathematically, the tradeoff of selecting this sorting algorithm is the complicated implementation. Quicksort required two additional helper functions and runs recursively. This makes it slightly more complicated to understand and implement. While the other sorting algorithms are mathematically slower, they are only a few milliseconds slower and offer simpler and more intuitive implementations. However, it is important to note that when using large data sets there will be a difference in terms of run time and if efficiency is something one is after then they should take the time to implement Quicksort. If one simply wants to sort a small set of data then any of the other sorting algorithms will be just fine.