An Empirical Study of Sorting Algorithms

V.M. Arroyo

Chapman University, Orange CA

Through Assignment 6, I was able to learn a lot about the sorting algorithms we learned about in class and how fast they run in the real world. In this assignment, we were required to implement the Quick Sort, Insertion Sort, and Bubble Sor algorithms. Since we had a choice for the last algorithm to implement, I also decided to implement the Selection Sort algorithm so I could cover all the Brute Force Algorithms.

To begin, I noticed that the Bubble Sort algorithm was significantly slower than all the other algorithms - including the other brute force algorithms. I found this interesting because we can sometimes see Bubble Sort and the other Brute force algorithms as having similar run-times, but evidently this wasn't the case. This may be because of the simplistic primitivity of the algorithm.

Selection sort performed consistently better than Bubble Sort, but never better than Insertion Sort. This, of course, is expected because Insertion can bypass a while loop for each pair of pre-sorted data, while selection sort is subject to an O(n*n) run-time even in the best case scenarios. So, these results were expected.

Quick sort and insertion sort battled consistently for the fastest algorithm, which seem to differ based on input. When I entered a small array or an array that was partially pre-sorted, I found that Insertion sort actually performed better than quick sort because the run-time was about O(n+i) compared to O(nlog(n)) for quick-sort. However, when I gave my program a large array with very little inversions, quick-sort performed significantly better than insertion sort. I found this interesting because this is what we would predict based on big O, but it was powerful to see it play out in real life.

Above all, this assignment served as a drastic testament to the power of Big O to predict relative run-time speeds for algorithms. Even though empirical analysis is more accurate because it takes other factors into account, it is also somewhat inconsistent - I saw different run-times for the same input with the same sorting algorithms. I do also find it interesting that bubble-sort was much slower than selection-sort, which also may be a system-dependent observation. All in all, this assignment made me truly appreciate the power of Big-O to predict run-time trends for complicated algorithms.