Assignment 6: Sorting Algorithms

Nick Lai - ID: 2282417

May 18, 2018

1 Overview

Assignment 6 tasked us to implement the sorting algorithms BubbleSort, InsertSort, QuickSort, and a sorting algorithm of our choosing and test their speeds. For the algorithm of choice, I picked BogoSort because it seemed like the most convoluted for no apparent reason. Speeds were tested based on the time it would take for each sorting algorithm to sort a randomly generated list of 10,000 values.

2 Results

The results from testing these algorithms were as expected, and the times each algorithm took on average showed the differences between the algorithms. BubbleSort, while easy to implement since it only requires a swap function, was the second slowest of the sorting algorithms implemented (only beating BogoSort). It achieved an average time of 565 ms to sort the randomized list.

InsertSort performs best on data sets that are already partially sorted, as it works by continuously moving lower values towards the left. It took an average of 515 ms to sort the randomized list.

QuickSort performed the best on this specific set of data, because it performs best on large samples of data. The only time values received were either an instant sort (less than 1 ms) or 15 ms, so it is safe to say it is the fastest for the specific situation presented.

BogoSort performed utterly terribly, as it has a runtime of O(n!). It never finished once, and I left the code running for a few hours just to see if it would ever finish. This is because the way BogoSort works is that it takes the list and, if it is not sorted, randomizes it until it is sorted. At 10,000 values, it would be working at a runtime of O(10,000!), which means that it will probably not be able to finish sorting this data set during our lifetime.