

Ben Bartol
Assignment 6
CPSC 350
Rene German
12/19/2020

Assignment 6 Results

While the five algorithms varied in efficiency, the results I encountered were to be expected. I figured that with a text file of numbers not too large, the difference would be completely unnoticeable, which it was, considering the recorded times to sort the numbers only varied in the thousandths of even milliseconds. But when referring to a file containing tens of thousands of numbers, the differences in efficiencies really tend to stand out.

Generally speaking, the quick sort seemed to be the most reliable and time efficient sorting algorithm. Its “quick” ability to sort the data comes through dividing and conquering, where the data is split up and pivoted around a center point. Although reliable, there are sometimes other sorting algorithms that may make more sense to use in certain situations as opposed to quick sort. In some circumstances, where the data might at least be *partially* sorted, the Insertion sort method would be the most ideal. But in this circumstance, this was not the case as it was extremely inefficient to sort, considering this method reads all of the data before moving a value. If time is not of the essence in our circumstance, then the bubble sort algorithm would also be a very appropriate option, where every number in the data is compared to its adjacent neighbor in the data, and swapping if the order is incorrect. It also takes a long time with a large set of data because not every time there’s an evaluation, there’s a swap.

