Jason Freedman

December 4, 2015

On my laptop, I created programs that randomly generated 100,000 numbers so I could see how long it took each of the sorting algorithms to sort through them. It took the insertion sort 8 seconds, the quick sort 28 seconds, and the selection sort 12 seconds. While I am not surprised that insertion sort was the fastest while quick sort was the slowest (which is ironic considering the name), I was rather surprised to see how large the difference between selection sort and quick sort was. The code I used for selection sort was about the same length as the code I used for quick sort (and both were much shorter than the code I used for insertion sort).

Based on the length of time, I’m assuming that quick sort (or at the very least, the version of it I used) is either more accurate than insertion sort and selection sort, or it is simply less efficient.

I am unsure how my choice of c++ affected the results of the comparison. Possibly it effected the time each one took if other languages have code that can run the sorting algorithms faster than c++ could.

Shortcomings of this empirical analysis are that it is expensive and, as shown by the results, time consuming when used on a large scale.