

Assignment 6 Analysis

Jon Flees

05/17/19

The sorting algorithms each have their own pros and cons. When it came to large data size, QuickSort was clearly the best. In fact, QuickSort can have an average run time of $O(\log n)$. This compared to the other algorithms can be significant when sorting lots of data. Bubble Sort, Insertion Sort, and my choice algorithm (Selection Sort) have run times of $O(n^2)$.

Interestingly, all the algorithms have a Big O of $O(n^2)$.

However, the times could (and did) vary significantly. Therefore, empirical analysis is not always the best choice for measuring one algorithm over another. For my program, I randomly generated doubles in the range of 0.1 to 99.9. I did this for data sets of 100, 1000, 10000, and 100000.

1 Bubble Sort

Bubble Sort's time to sort was fastest for 100 doubles, yet significantly slow for 100000. Run Times: 100 = .000304 seconds 1000 = .00258 seconds 10000 = .2530 seconds 100000 = 29.2222 seconds

2 Insertion Sort

Insertion Sort's times were consistently between the best and worst for each data quantity.

Run Times: 100 = .000352 seconds 1000 = .00132 seconds 10000 = .0726 seconds 100000 = 6.5312 seconds

3 Selection Sort

Selection Sort was the third best algorithm for each quantity other than 100, which it was actually the slowest.

Run Times: 100 = .000427 seconds 1000 = .00218 seconds 10000 = .1094 seconds 100000 = 9.9305 seconds

4 QuickSort

QuickSort's times were significantly better than the other algorithms as the amount of data increased. By far, the best algorithm for lots of data.

Run Times: 100 = .000305 seconds 1000 = .00099 seconds 10000 = .0054 seconds 100000 = .0385 seconds