CPSC 350 Assignment 6 Report

Irene Liu

I. Introduction

I performed an empirical analysis of the four following sorting algorithms:

- Bubble sort
- Insertion Sort
- Quick Sort
- Gnome Sort

II. Results

With an initial input size n = 30000, these were the results: Bubble sort: 9.85s, insertion sort: 2.19, quick sort: 9ms, gnome sort, 5.4s. I then increased the input size to 50000 and noticed a large change in the time spent with the bubble sort. After 3 runs with 50000, these were the average run times: bubble sort: 27s, insertion sort: 6.0, quick sort: 15ms, and gnome sort: 14s. Finally, I used an input size of 110000, and recorded these times: bubble sort:61s, insertion sort: 14s, quick sort: 18ms, gnome sort: 34s.

CPU usage was fairly linear across all 4 algorithms. With n = 11000, it always hovered around 20%.

III. Conclusion

These results are consistent with what we've learned about sorting algorithms. The bubble sort has the longest run time because there are no conditions under which is can exit prematurely. In comparison, the insertion sort and gnome sort have better average run times in cases where the data is partially sorted. The growth rate of the merge sort is minimal between 30000 and 50000, which would indicate a linear logarithmic growth rate.

It also makes sense that there was little variation for CPU usage between each algorithm, since most of them are in-place sorting algorithms with the exception of merge sort. However, merge sort takes a minuscule amount of time in comparison, and I still didn't notice any spikes in CPU usage.