

# Assignment 6

Aubrey Fernando

May 13, 2018

## I. INTRODUCTION

In Assignment 6, I implemented four different sorting algorithms: Quick Sort, Bubble Sort, Insertion Sort. The last algorithm I chose to implement was Shell Sort.

## II. RESULTS

The times it had taken the sorting algorithms to sort a file of number were relatively different from each other. I had tested all four sorting algorithms against files that had contained anywhere from 8 to 5000 numbers. The results are from the times it had taken the sorting algorithms to sort a file of 5000 numbers.

Quick Sort had taken the least amount of time with small and large files, while Bubble Sort had taken the longest in both cases as well. Quick sort is the fastest out of the sorting algorithms tested, especially with a large amount of data to sort through because it is able to successfully break the sorting process into two parts by partitioning the data based on pivots. Bubble Sort is able to get the job done sorting the data correctly, but not the most efficiently, so it's best to use Bubble Sort when you are dealing with smaller sets of data.

Insertion sort performed moderately well and would be a good option if the data was already partially sorted, allowing less swaps and copying to take place. My choice of sorting algorithm, Shell Sort, is a variation of Insertion Sort. Instead in Shell Sort data can be exchanged for items that are farther away from each other. Shell Sort's performance was a little better than Insertion Sort, and this might because Insertion Sort moves times one position at a time and does sorts through by many movements.

Testing the sorting algorithms in C++ is a good option because C++ can easily be used to implement these algorithms, but other programming languages like Java or Python could definitely have an influence on performance.

Some shortcomings of this analysis is that the way we had used to measure the time each sorting algorithm had taken is not the most accurate way to test how long each had taken. Another shortcoming is that we are only testing on a few data sets, we could later possibly test each sorting algorithm with data sets that we expect should perform well.