

# Assignment 6

Jack McGrath

December 15, 2018

## Abstract

LaTeX file for Assignment 6

## 1 Time Differences

I was actually somewhat surprised on the time differences for the different algorithms. Since I outputted all of my run-times in microseconds I was able to see minute time differences when even running small data sets. As expected Bubble Sort was the slowest. However, on this small of a Data Set Quick Sort was actually slightly longer than Insertion Sort by 125 microseconds, on a data set of 25 numbers. However the data set was partially sorted when coming into the algorithm, giving a slight advantage to the runtime of Insertion Sort.

## 2 Tradeoffs

When picking one algorithm over another there are several factors to consider. Foremost is the actual Big-Oh of the different sorting algorithms. Generally an algorithm with a small big-oh is going to be the best to implement, but the tradeoff for this is it can either be more complex to implement, or be bad on memory, like Merge Sort. However sometimes using something that's faster isn't all that necessary. For example if you were working with a very small data set, and you didn't have a lot of memory to spare in your computer, using an algorithm like Bubble Sort or Insertion Sort might make more sense.

## 3 Shortcomings of this Empirical Analysis

Since we're working within the bounds of our own computer its hard to test these algorithms to there extremes using very large data sets. Its also hard to be super accurate with our time or memory consumption as we aren't running a separate application to see our real runtimes and memory used from our code at run-time.