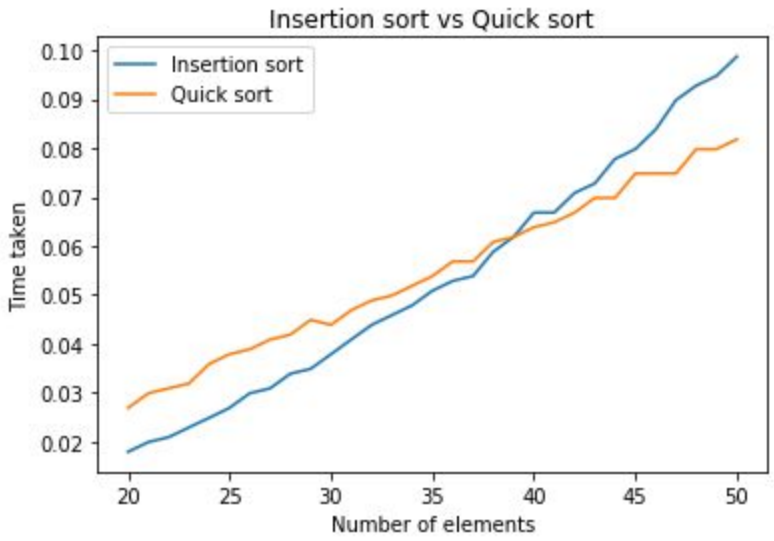
Question 1: Quick sort vs Insertion sort

- *Hypothesis*:
  We know that quick sort is an O(n log n) algorithm and insertion sort is O(n^2). Because of this, we expect for quick sort to be faster as n approaches infinity. However, we guess that for small n, insertion sort will be faster. We believe that n=10 will be the crossing point for where insertion sort is no longer faster than quick sort.
- *Methods*:
  We generated arrays of sizes starting at 1 until X. We filled these arrays with randomly generated numbers. Then, because sorting small arrays can take extremely short amounts of time (near zero), we sorted the same array sizes many times in order to produce a more accurate time. We then plotted the data.
- *Results*:
  We ran various experiments with different source code, and ended up concluding that the



  value for n that the two sorting algorithms cross at n=40.
- *Discussion*:
  This value of n that we found was larger than we thought it was going to be. We did not run into many struggles, other than generating concise and clear graphs.
- *Conclusions*:
  Under the conditions tested, insertion sort is a faster sorting algorithm for $n < 39$, while quicksort is faster for $n >= 40$.