Comparisons Between The Sorts

- Identify the respective time complexity for each sort and include what you have to say about the constant.

We will identify the average time complexity of each sort and compare them with each other.

Going with quick sort first, the average time complexity is O(nlogn), as stated on the lecture slides as well, by looking at the partition and how it splits the elements in one set which will take O(n/9) and O(9n/10) in other set, the total equation to calculate the time complexity comes down to $O(n/9) + O(9n/10) + O(n)$ with O(n) coming from the recursive loop. This addition of the complexities from the recursive loop and partition diverges the time complexity to O(nlogn).

For shell sort, the average time complexity depends on the size of the gap sequence. But from the lecture, I believe the professor mention a prediction of about O(n^5/3) judging from most real-world cases we would have to use this for so let's go with that.

Bubble sort has an average time complexity of O(n^2), simply seen from the two loops comparing adjacent values.

Binary insertion sort has an average time complexity of O(n^2) because despite using binary search to bring down comparison value, the series of swaps required generally remains the same.

So, in comparing all the average time complexities of all sorts, we can reasonably conclude that for small sets like those that have 50 elements, using sorts like Bubble sorts or binary sorts might be better since they would take less time compared to quicksort. In average terms though, quicksort might be better to use since often datasets will not be under 50 elements and quicksort is on average much faster than other sorts. Its quick speed is reflected in how its # of moves and comparisons are relatively lower than other sorts.

- What you learned from the different sorting algorithms.

The major lesson from this assignment is that different sorting algorithms are best for different situations and depending on the size of data and data type. If you have small data, it would be perhaps be better to use 'worse' sorting like bubble sort, else it would be better to use sorts like shell sort or quicksort.

- How you experimented with the sorts.

I measured how many moves and comparisons each sort had to make to sort through the same data.