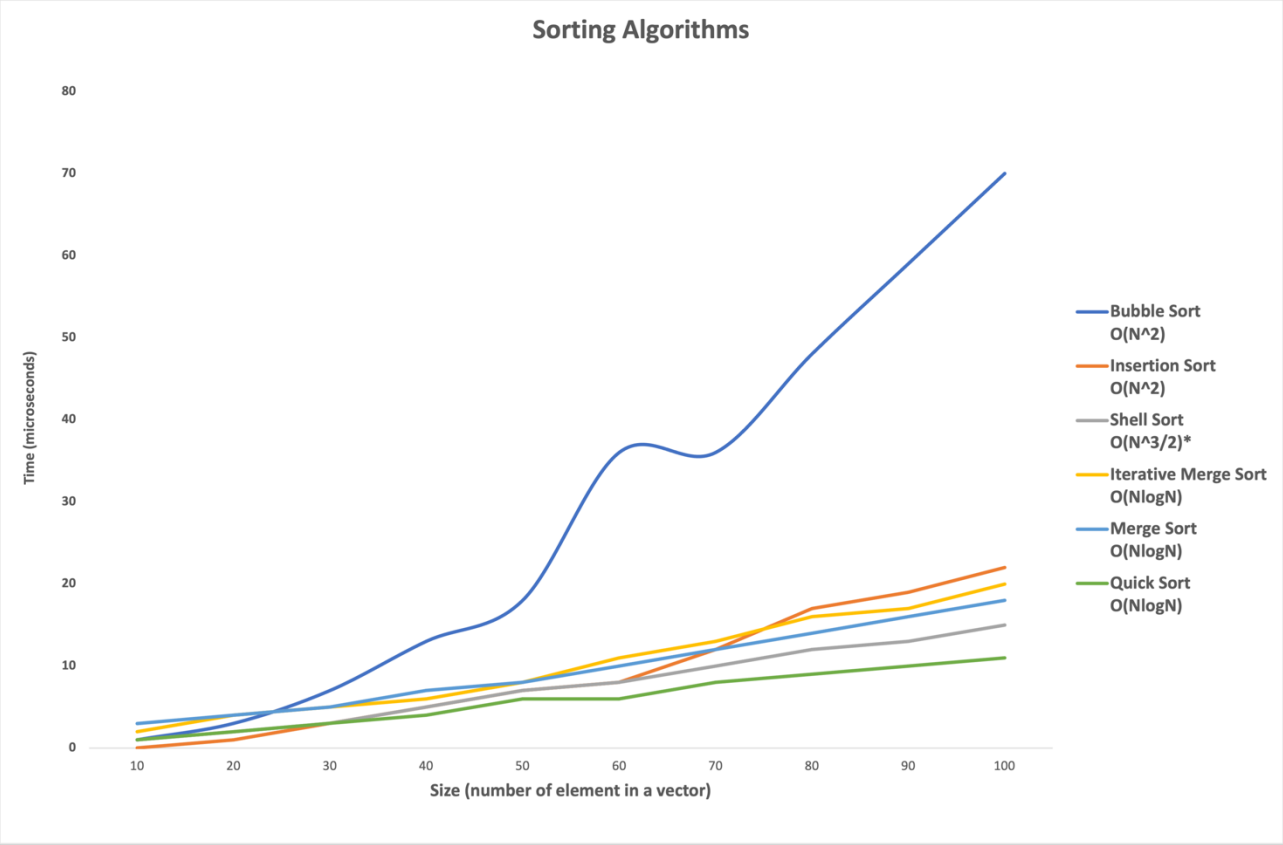
Actual Performance of Sorting Algorithms

There are different sorting algorithms available to arrange elements in a container. This report will discuss six sorting algorithms: Bubble Sort, Insertion Sort, Shell Sort, Quick Sort, Iterative Merge Sort, and Merge Sort. This report will show their similarities in terms of time complexity and demonstrate actual performance of these algorithms on vectors containing integers in with different sizes. It will also compare how algorithms with same or similar time complexity differ in actual performance.

Figure 1 (below) shows all six sorting algorithms for small vectors of sizes 10 to 100. Even though the Bubble Sort and Insertion Sort have same time complexity of $O(N^2)$, there is a difference in their actual performance. On smaller data sets, Insertion Sort performs like Iterative Merge Sort, Merge Sort, Quick Sort, which have time complexity of O(NlogN), and Shell Sort $O(N^{3/2})$[1]. Further, this report will demonstrate the actual performance difference in all six sorting algorithms in bigger data sets. It will compare Merge Sort and Insertion sort since they have same time complexity and the remaining four algorithms will be compared together.

Figure 1



[1] Note: Shell Sort performance varies in terms of time complexity as it was discussed in lecture on 16 November 2022.

Figure 2 (below) demonstrates Bubble Sort and Insertion Sort's actual performance. Both sorting algorithms have same time complexity of $O(N^2)$, yet their actual performance varied. As the size of a vector got bigger, the Insertion Sort performed much better than Bubble Sort. Additionally, both Insertion Sort and Bubble Sort's overall performance decreased starting with vector size of 45,000 elements. Overall, Bubble Sort for the vectors with sizes from 1,000 to 100,000 on average took about 2.9 times more time relative to Insertion Sort's time.

Figure 2

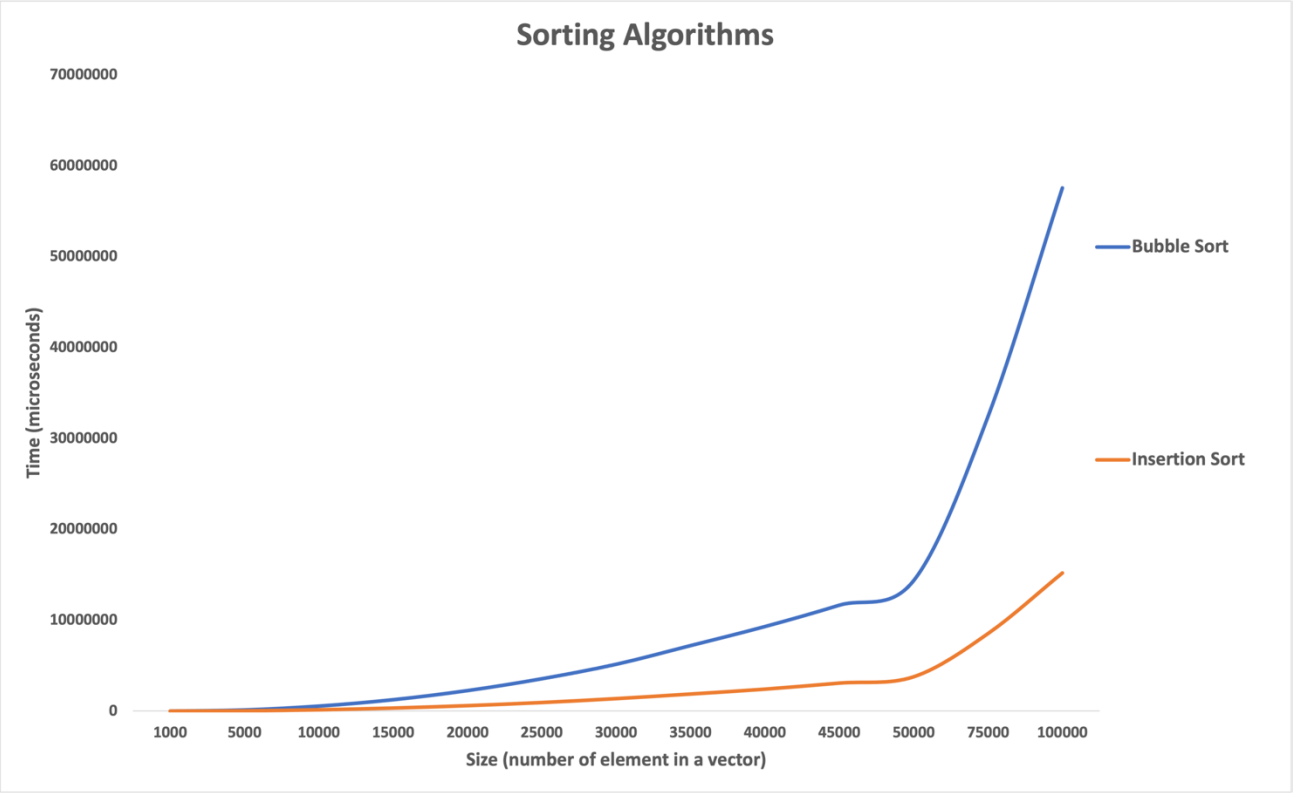
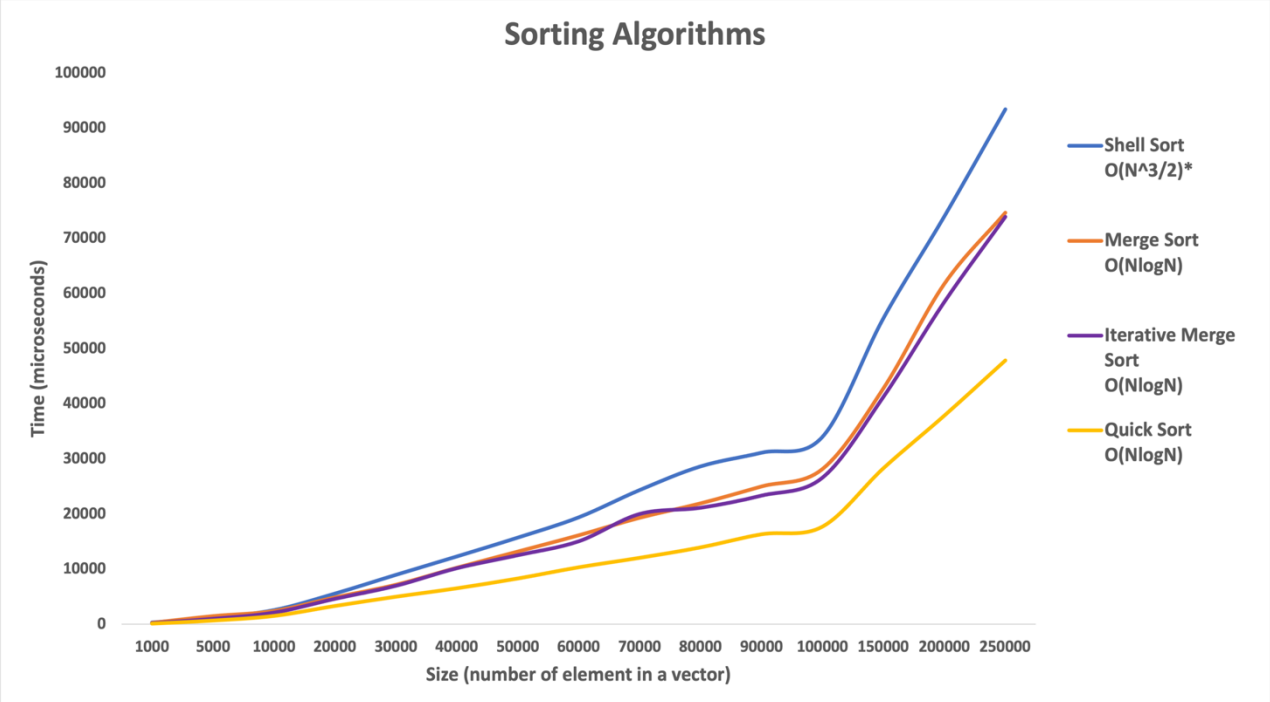


Figure 3 (next page) shows the actual performance of Iterative Merge Sort, Merge Sort, Quick Sort which all have a time complexity of O(NlogN), and Shell Sort $O(N^{3/2})$. Even though Iterative Merge Sort, Merge Sort, Quick Sort same time complexity, actual performance of these algorithms varies. Although Merge Sort and Iterative Merge Sort have similar divide and conquer approach, on average, Merge Sort took 1.1 times more time relative to Iterative Merge Sort's time.

Quick Sort was the fastest among 4 sorting algorithms. Relative to Quick Sort's time, on average, Shell Sort took 1.9 times more, Merge Sort took 1.6 times more, and Iterative Merge Sort took 1.5 times more to sort the vectors with sizes from 1,000 to 250,000 elements.

Figure 3



In conclusion, sorting algorithms with same time complexity differ in actual performance. Overall, Quick Sort was one of the fastest sorting algorithms among the six sorting algorithms discussed for smaller and larger data sets. Bubble Sort's performance relative to Insertion Sort decreased as the data size increased. Also, Iterative Merge Sort would be a better alternative to Merge Sort since it takes less space in terms storage and it is a little faster than Merge Sort.