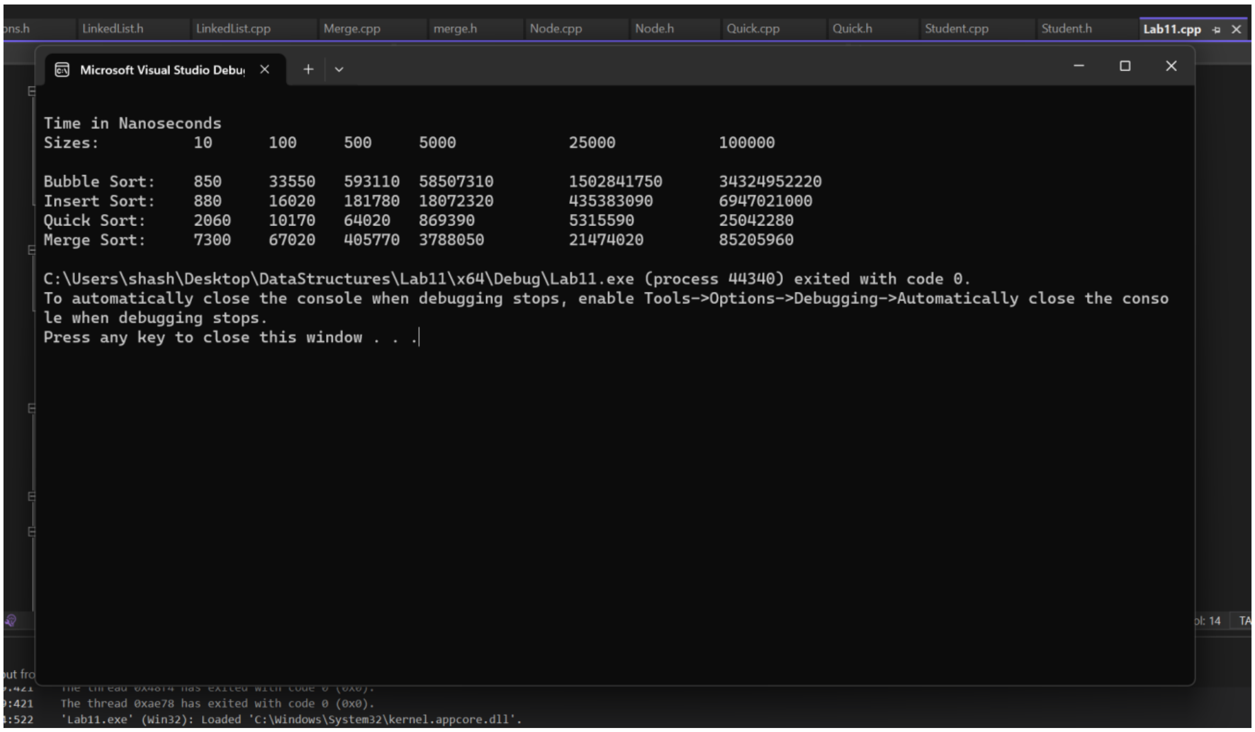
Lab 11 Report

The objectives and concepts explored in this lab assignment include various fundamental sorting algorithms such as bubble sort, merge sort, quick sort, and insertion sort. We also take a look at the performance of these sorting algorithms using the Big O notation. These concepts are fundamental for a software engineer and are used a lot in the software development industry.

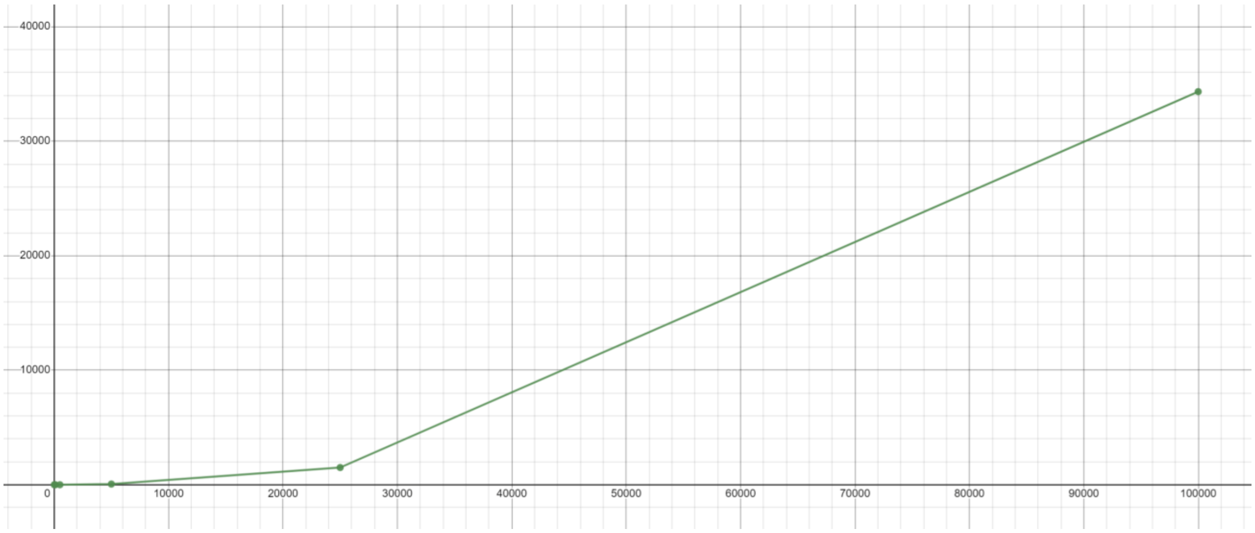
Output:



Output Table in ms :

	10	100	500	5000	25000	100000
Bubble Sort	0.00085	0.03355	0.59311	58.50731	1502.84175	34324.95222
Insertion Sort	0.00088	0.01602	0.18178	18.07232	435.3830	6947.021
Quick Sort	0.00206	0.01017	0.06402	0.869390	5.31559	25.04228
Merge Sort	0.0073	0.06702	0.40577	3.78805	21.47402	85.20596

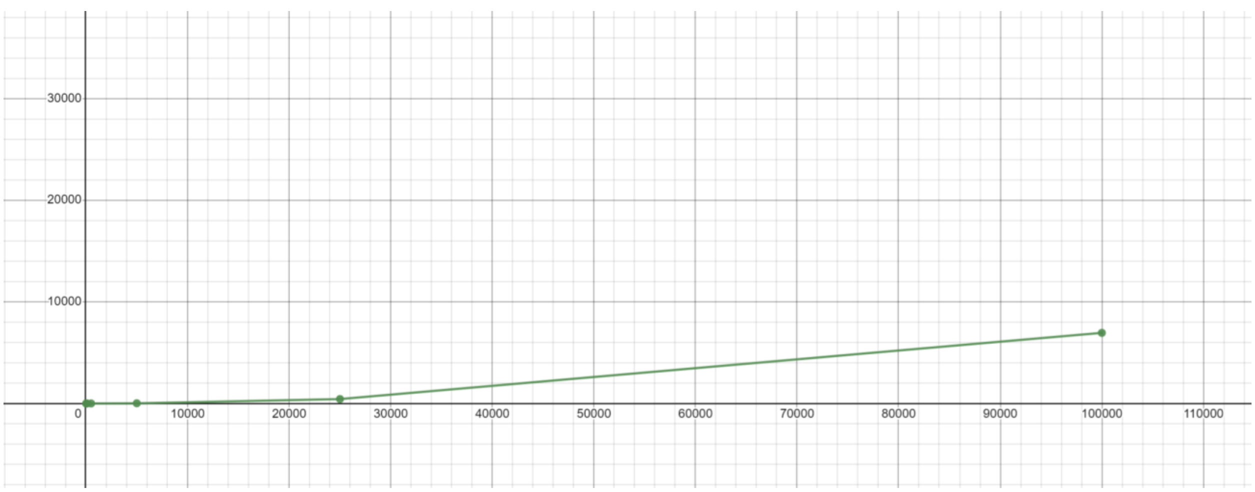
Bubble Sort:



For the bubble sort algorithm, the big O notation is O(n^2)

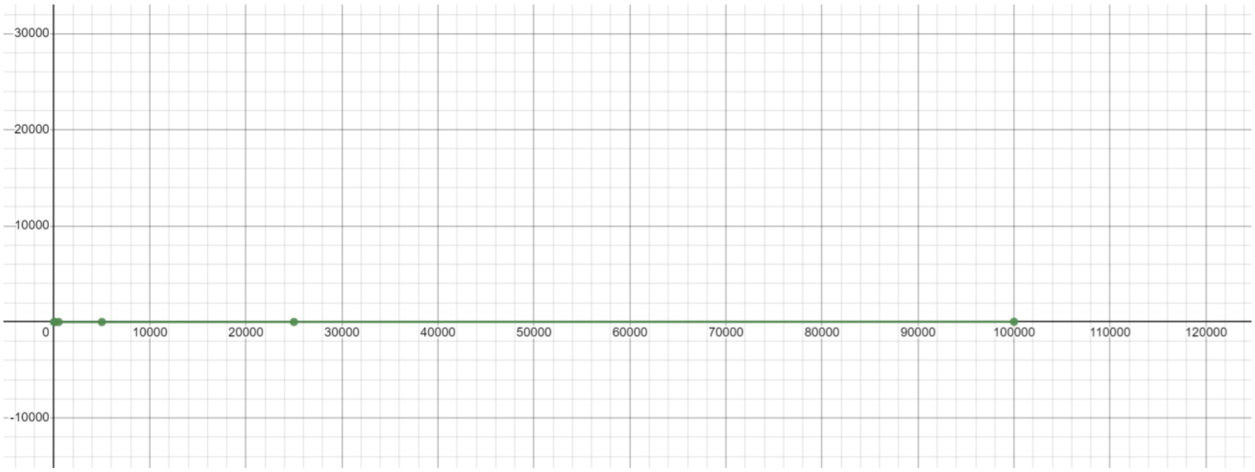
As seen in the graph, it forms a parabolic curve which shows the n^2 in O notation. We observe that it was the fastest for array size = 10 but as we add more elements or increase the size of the array, the performance decreases. It matches well with what I expected as it was very quick for smaller array size and as soon as array size increases, it takes more time.

Insertion Sort:



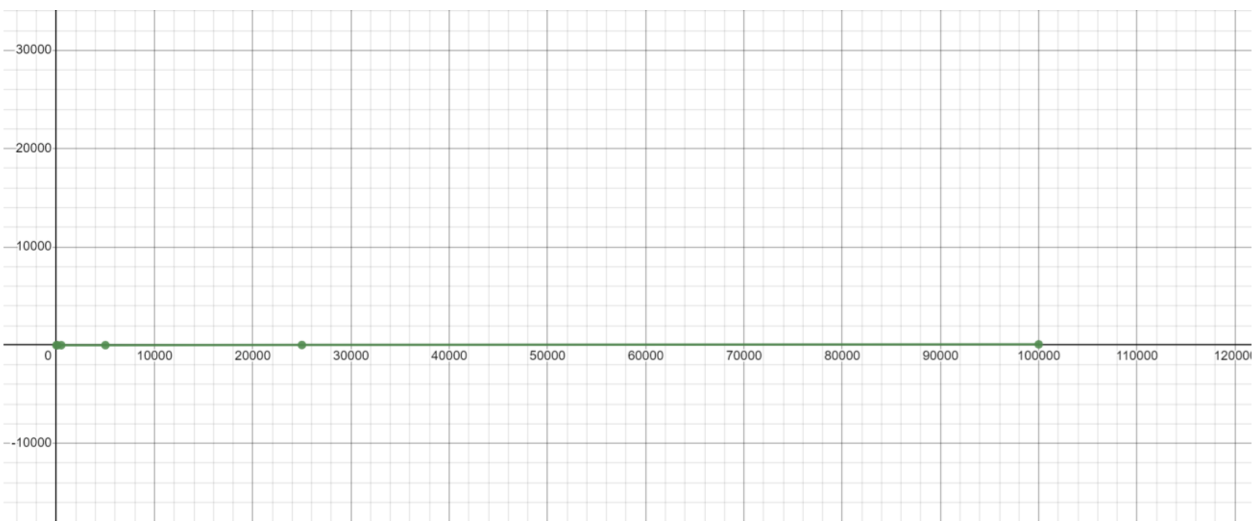
For the Insertion sort algorithm, the big O notation is also O(n^2). As seen in the graph, it also forms a parabolic curve. As mentioned in class, it is 2 to 3 times faster than the bubble sort algorithm. As expected it is very quick when the array size is small. But as the array size increases, the performance decreases. However, it is still faster than bubble sort when the array size is larger.

Quick Sort



Quick Sort has O(nlogn) in average or worst case. The quick sort algorithm had the best performance overall as the size of the array increases. The performance was very close to merge sort but if we look at the graph carefully, quick sort is a bit quicker. The graph is closer to O(1) rather O(nlogn) which was not expected by us. It could form a nlogn curve if we add more elements and increase the array size

Merge Sort



For merge sort, the Big O notation is O(n log n)

The merge sort algorithm was faster than expected and the graph shows the performance is a bit faster than a nlogn graph. Merge sort uses recursive calls which uses a lot of memory. This made us believe that the performance will not be this fast. The performance exceeded our expectations in this case.