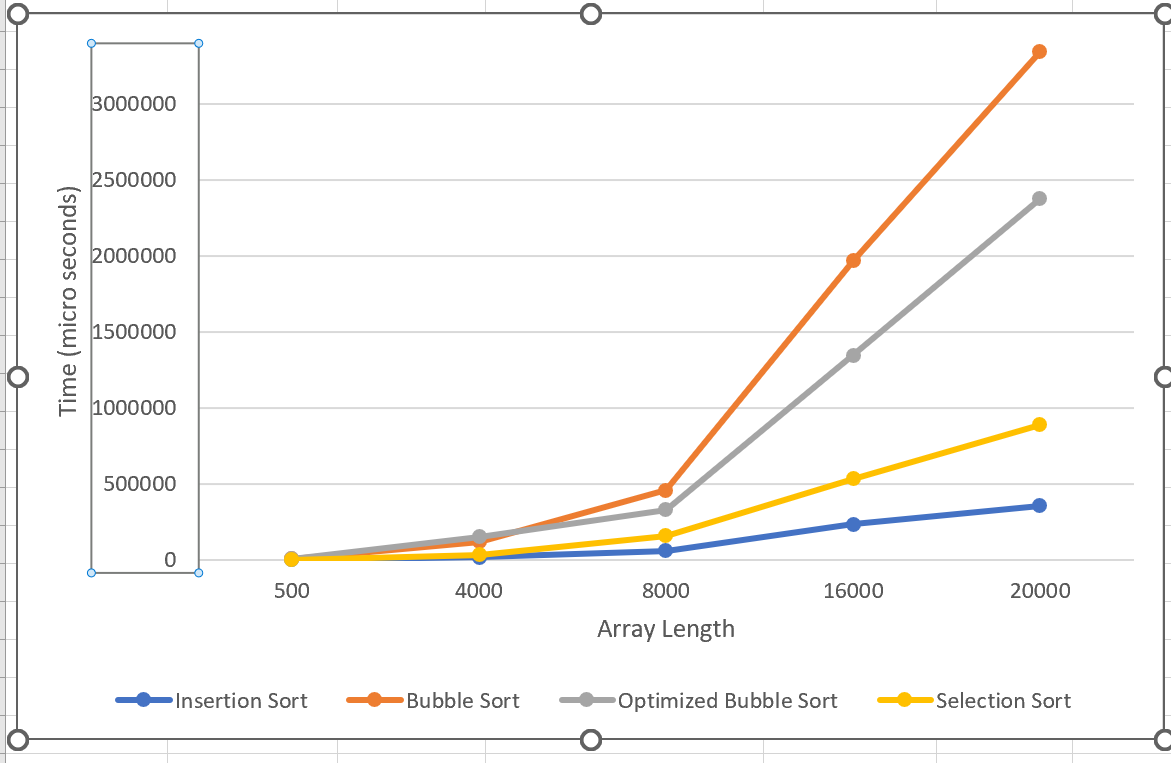
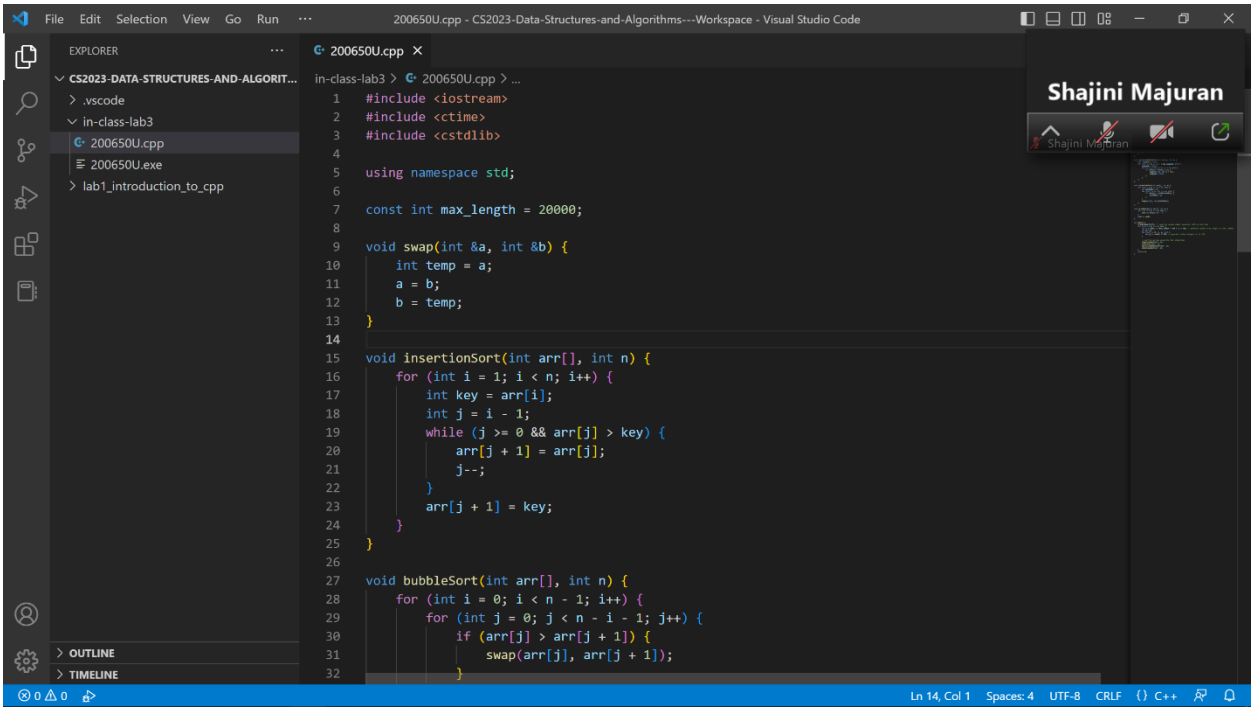
Array Length	Insertion Sort	Bubble Sort	Optimized Bubble Sort	Selection Sort
500	234	1695	942	386
4000	13432	113799	149977	30684
8000	54092	454696	325195	155337
16000	230736	1963460	1344066	530181
20000	355160	3340931	2374208	884604





Insertion sort:

Best case: O(n)

Worst case: O(n^2)

Average case: O(n^2)

Bubble sort:

Best case: O(n)

Worst case: O(n^2)

Average case: O(n^2)

Optimized bubble sort:

Best case: O(n)

Worst case: O(n^2)

Average case: O(n^2)

Selection sort:

Best case: O(n^2)

Worst case: O(n^2)

Average case: O(n^2)

Note that the best-case time complexity is when the input array is already sorted, and the worst case time complexity is when the input array is in reverse order. The average case time complexity approximates the time complexity for random inputs.

Again note that from the above graph, Insertion sort performs best and Bubble sort performs worst.

Among the four sorting algorithms mentioned, Insertion sort has the best-case time complexity of O(n) which means that if the input array is already sorted or almost sorted, insertion sort will perform optimally. Bubble sort has the same best-case time complexity as insertion sort, but its worst-case and average-case time complexities are O(n^2), making it inefficient for large arrays. Optimized bubble sort is a variation of bubble sort with a slight optimization, but it still has the same time complexity as bubble sort in both the worst-case and average-case scenarios.