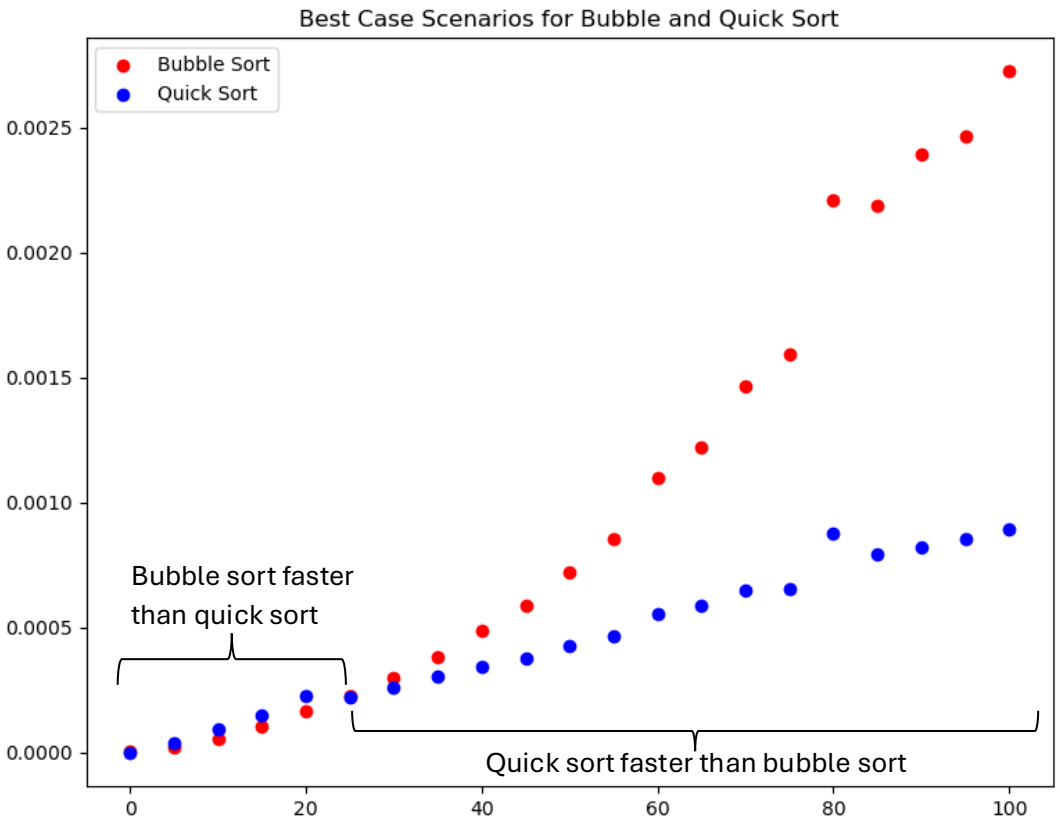
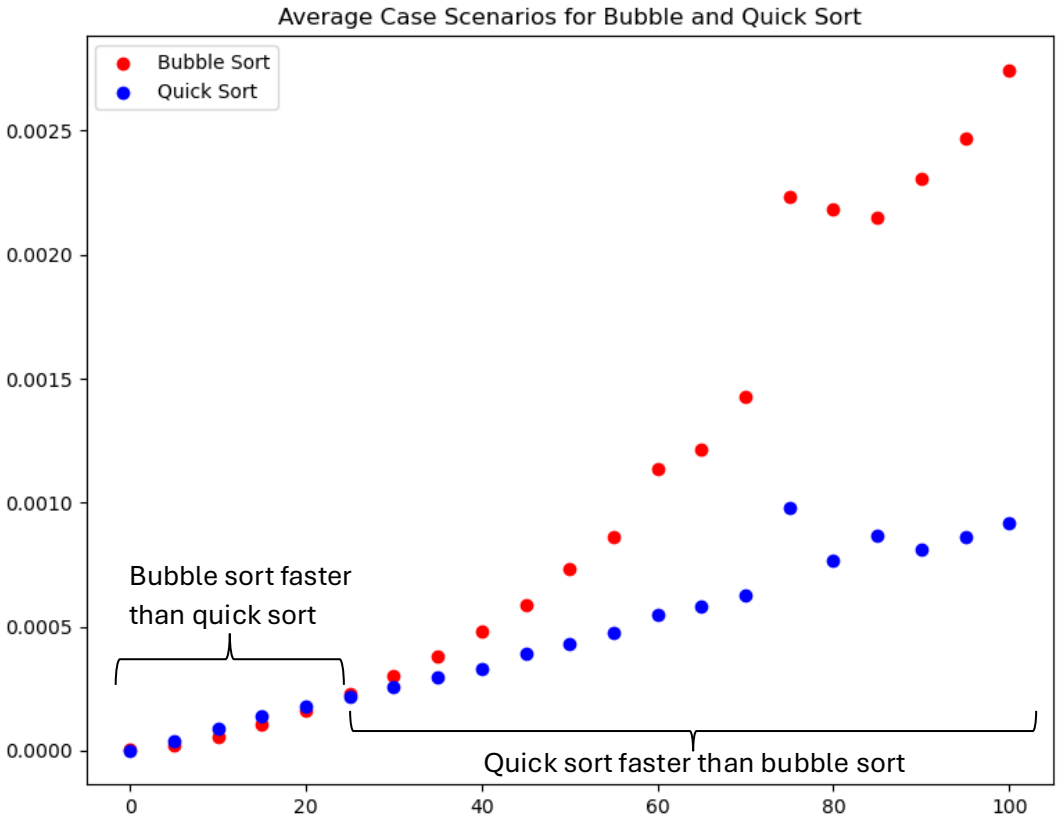
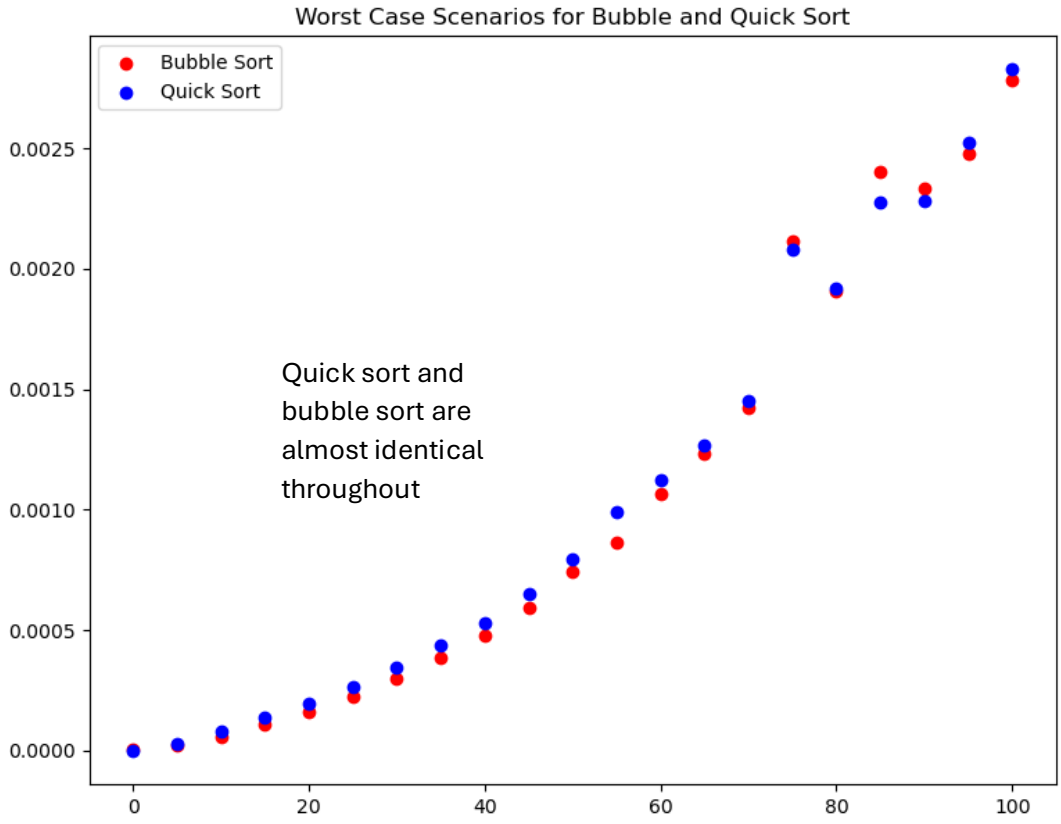
ENSF 338 Lab 3 Exercise 2

Question 3:




Average Case Scenarios for Bubble and Quick Sort
Bubble Sort
Quick Sort
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
0
20
40
60
80
100
Bubble sort faster
than quick sort
Quick sort faster than bubble sort



Question 4:

The input size threshold that determines whether the input is small or not is an input of size 25. From the plots for both the best case and average case scenarios, it is evident that quick sort begins to out-perform bubble sort at an array size of around 25 elements. Both of these scenarios implement quicksort with the middle element as the pivot. In the best-case scenario, the array is partitioned evenly throughout the sorting process, whereas in the average-case, theoretically it should not matter which element you choose as a pivot. Since the array is shuffled, any element should theoretically partition the array roughly evenly; there may be some uneven partitions in the sorting process but in general, most partitions should be roughly even by the end of the process.

In the worst-case scenario, it is evident that quicksort and bubble sort are similar throughout the entire sorting process. This is to be expected since in the worst-case scenario, both bubble sort and quick sort have a quadratic time complexity $O(n^2)$. In the worst-case, bubble sort would have to move every element to its opposite side, meaning that the original input is sorted in reverse. Comparatively, quick sort would partition the array in such a way that it results in one subarray having a size of one, and the other subarray containing all other elements. Therefore, it would have to move every element into either the left or the right subarray. From the above graphs we can see that the demonstrated input size threshold is 25 elements.