1. Algorithm Analysis (18 points)
For each algorithm listed, you are required to identify the asymptotic run times in the worst-case, average-case, and best-case scenarios (1 point each). Additionally, you must specify the types of data vectors (e.g., sorted, reversed) that correspond to these runtime bounds (1 point each).

Algorithm	Best-case	Average-case	Worst-case
Bubble sort:	O(n) Sorted vector	O(n^2) Randomly sorted vector	O(n^2) Reverse sorted vector
Insertion sort:	O(n) Sorted vector	O(n^2) Randomly sorted vector	O(n^2) Reverse sorted vector
Selection sort:	O(n^2) Sorted vector	O(n^2) Randomly sorted vector	O(n^2) Reverse sorted vector
Quicksort:	O(nlog(n)) Sorted vector	O(nlog(n)) Randomly sorted vector	O(nlog(n)) Reverse sorted vector

3. Analysis of Generated Graphs (7 points)
For each graph produced in question 2, describe your observations. Consider whether the curves match expected patterns and whether some algorithms perform better than others under certain conditions. Note any unexpected results. Reference your findings in response to question 1 for a comprehensive analysis.

For bubble sort, and insertion, in best case scenario they had nearly identical runtimes which resemble the growth rate of O(n). Then with quicksort being the slowest and selection coming in second. Which correlates with the table from problem 1. I noticed that in the best case scenario either bubble or insertion sort would work best.

For average case buble sort directly correlated with the table in one with the growth rate being n^2. For Insertion sort there was a discrepancy. This could be caused by numerous things such as how the code is compiled and how cores are being utilized on my system while running the code. But both insertion and selection sort both show us that the times are infact squared just not as slow as bubble sort. Qucksort on the other hand seems to be the fastest as the growth rate is the least.

For worst case bubble sort seemed to be faster than average times but still slower than all the other sorting methods. Quick sorts seemed to run slower than insertion and selection sort as expected from what we collected from the table. With Insertion and selection sort we have almost identical runtimes which is expected if we were to compare our tables.