**Assessment 7**

Results for the **extraLargeArray**

* insert 935.9591 ms
* append 3.0389 ms

Results for the **largeArray**

* insert 9.5365 ms
* append 698.5 μs

Results for the **mediumArray**

* insert 205.8 μs
* append 172.8 μs

Results for the **smallArray**

* insert 54.8 μs
* append 104.6 μs

Results for the **tinyArray**

* insert 37.3 μs
* append 90.5 μs

A graph with different colored lines

Description automatically generated with medium confidence

A graph with a line and text

Description automatically generated with medium confidence

A colorful circle with a number

Description automatically generated

From the graphs I provided above it becomes clear that the execution times of the functions (Insert and Append) increases as the size of the array increases but at very different rates. The insert function shows a consistent execution time as the array grows indicating this would be linear scaling. The append function is very different. It seems to show that regardless of the size of the array, the execution times are very similar. After some research, I believe that the insert function becomes so slow compared to the rest because it must shift elements before adding the new element in the array. As the array grows, the number of elements that must be shifted grows linearly so very large arrays become “slow” compared to the insert function.