Module 7 Project

Results	insert	append
tinyArray	36.8 μs	90.3 μs
smallArray	63.5 μs	111.2 μs
mediumArray	215.5 μs	169 μs
largeArray	10.9021 ms	771.7 μs
extraLargeArray	1.0664586 s	2.5724 ms

From the table, the observation is that there is a pattern in the execution times for the two functions. It is evident for both functions that as the array size increases, the run time also increases. Despite both functions executing at fast speeds, the increase rate for both functions differs and thus showing the differences in scalability. The increase rate for the insert function is greater than the append function as we see the run times between the array sizes is significantly increases in comparison to the append function. This shows that the append function scales better and is also a faster function overall.

Extra Credit:

The reason why the slower function is slower is because it using the unshift method while the faster funciton uses the push method. The unshift method is slower than the push method because the push method only has to add an element to the end of an array. While the unshift method, which adds elements to the begining of an array, must shift every exisiting element in the array in order to accommodate elements added to the beginning of the array. This extra step takes time and increase in time depending on the numer of elements that need to be shifted prior giving a time complexity of O(n) while the push method is constant with a big O of O(1).