Assessment 7 Notes

1.) **Take note of the timing result for the extraLargeArray results– comparing when the extraLargeArray is passed to doublerAppend and doublerInsert.**

Results for the extraLargeArray:
insert 1.922029835 s (passed to **doublerInsert**)
append 5.611721 ms (passed to **doublerAppend**)

Results for the tinyArray:
insert 128.401 µs
append 123.87 µs

Results for the smallArray:
insert 75.051 µs
append 165.557 µs

Results for the mediumArray:
insert 266.187 µs
append 230.457 µs

Results for the largeArray:
insert 30.49204 ms
append 780.374 µs

ARRAY	INSERT	APPEND
tinyArray	128.401 µs	123.87 µs
smallArray	75.051 µs	165.557 µs
mediumArray	266.187 µs	230.457 µs
largeArray	30.49204 ms	780.374 µs
extraLargeArray	1.922029835 s	5.611721 ms

2.) **Read over the results, and write a paragraph that explains the pattern you see. How does each function "scale"? Which of the two functions scales better? How can you tell?**

As the tests progress, the bigger the arrays get. For example, we tested arrays ranging from tiny to extra large. When we changed the size of the array, the runtime matched that specific array size. As the array size got larger, the runtime also increased. After comparing the results, it is evident that the run times are faster with the doublerAppend method. I believe this is due to the different methods being used in each function. While the doublerAppend function uses the push method, the doublerInsert function uses the unshift method. Using unshift compared to push, results in a slower runtime.

3.) **For extra credit, do some review / research on why the slower function is so slow, and summarize the reasoning for this.**

The doublerInsert function is the slower function due to the method it uses. The push method is used in the doublerAppend function and has a runtime of O(1). Whereas, the doublerInsert function uses the unshift method, which has a runtime of O(n). The unshift method is slower because it is more complex compared to the push method. While the push method just adds onto the end of an array, the unshift method must increment all elements of the array. Therefore, all indexes in the array end up being modified. This results in the runtime being slower.