Unit 7 Assessment

Step 1:

3.) Runtime Table for Insert and Append methods:

	Tiny Array	Small Array	Med Array	Large Array	Ex. Lrg. Array
Insert	26.8 μs	36.7 μs	124.4 μs	6.4422 ms	689.212 ms
Append	68 μs	80.8 μs	103.6 μs	352.1 μs	2.1925 ms

4.) According to the results displayed on the table above, the Append method scales much better than the Insert method but the Insert method is better for small data sets. This is due to the way **.push()** (Append) and .**unshift()** (Insert) work in JavaScript. The append method adds the new element to the end of the list, so regardless of the length of the list it will simply add the new element without having to loop through the entire list, whereas the insert method adds the element to the beginning of the list assigning it the 0 index position and has to reassign the index value of all the other elements on the list, for which it has to loop through the entire list after inserting the new element at index 0. Because of this, the append method generally has a time complexity of O(1), and the insert method generally has a time complexity of O(n).

5.) The insert method is much slower because as the number of elements on the list increases, that means it has to recreate a list with the new element at index 0 and every other element on the original list will be added afterwards assigning them their new index value and this new list will then overwrite the old list. When you have to do this thousands or more times to add a single element to the beginning of the list, where the general Time Complexity of O(**n**) comes in, ***n*** being the number of elements on the list, this method stops being the most time efficient when compared to the append method.