Runtime Analysis:

Although both functions finished quickly, when looked at closely we can see that the doublerAppend function finished in mere milliseconds while the insert function took 1 whole second. Both times are short, but it's clear that the append function is quicker.

	tinyArray	smallArray	mediumArray	largeArray	extraLargeArray
Append	162.9 µs	103.6 µs	162.6 µs	514.4 µs	2.6~ ms
Insert	94.4 µs	47.4 µs	184.3 µs	9.1279 ms	1~ s

After running all the different array sizes the pattern, I notice the most is that the append function increases at a linear pace as the input is increased while the insert function starts to exponentially increase in runtime as the input grows.

The append function's scale is O(n) because it increases at a constant rate because it will only run one more time for every item that's added to the array.

The insert function's scale on the other hand is O(n^2) because it increases exponentially as the input is increased. This is due to the nature of the unshift method causing the computer to move every individual item in the new array over every time another item is added. This adds up as the input gets larger.