The results for doublerAppend were 3.6147 ms, while the results for doublerInsert were 1.643502 s.

	doublerInsert	doublerAppend
tinyArray	51 μs	124 μs
smallArray	63 μs	149 μs
mediumArray	268.3 μs	202.4 μs
largeArray	13.9033 ms	753.8 μs

Running the tiny and small arrays showed that the Insert function was more time efficient in the smaller functions. However, the medium, large, and extra large arrays all were slower for the Insert function. After doing some research, I found out that unshift, which is what the Insert function is using, has to accommodate every object in the array every time it inserts a new variable at the beginning of the array. This gives that function a time complexity of O(n). Push on the other hand doesn't have to read and accommodate the array every time it runs, it simply tacks on the new variable to the end of the array and continues with its loop giving it a time complexity of O(1).