Runtime

		doublerAppend (push)	**doublerInsert (unshift)**
	Size	Time	Time
tinyArray	10	5.3 µs	7.5 µs
smallArray	100	8.8 µs	14.6 µs
mediumArray	1000	37.8 µs	184.9 µs
largeArray	10000	128 µs	9.9527 ms
extraLargeArray	100000	2.3894 ms	896.1296 ms

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?

The DoubleAppender’s block of code is using the push method for assisting with editing its array function, which take any added item and place it at the back of an existing list, while the Double Insert uses the unshift method to place items in front. The DoubleAppender’s push method remained to perform faster than the Double Insert’s unshifted method.

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

The unshifted method is slower because it takes longer to add items into an existing list that acquires a specific order in which it is provided or given at the front of a list, especially an extremely long list. I believe it takes more effort and time it takes for a system to think when moving items to the front and right of a list when making changest to it. If the order of items in a long list is specific and using the unshifted method is crucial or important depending on the situation, the system also needs a longer time to process that action as well.