	tinArray		smallArray		mediumArray		largeArray		extraLargeArray	
	insert	append	insert	append	insert	append	insert	append	insert	append
runtime	19.042us	50.042us	27.917us	57.375us	125us.	87.292us	6.412917ms	386.459us	767.95175 ms	6.567333ms

doublerAppend scales linearly, doublerInsert scales quadratically. doublerAppend scales better as the array size increases

doublerInsert has the time complexity of O(n^2). A for loop has time complexity of O(n) and inside the for loop the unshift method has the time complexity of O(n) as well. Hence doublerInsert has the time complexity of O(n^2)

doublerAppend has the time complexity of O(n) , a for loop has O(n), and inside the for loop the push method has the time complexity of O(1). Hence doublerAppend has the time complexity of O(n)

From above method you can see doublerAppend method scale better than doublerInsert. Because the time complexity O(n) will perform better than O(n^2) when n increase to a larger number

The reason for doublerAppend scale better than doublerInsert is doublerAppend uses push method, doublerInsert uses unshift. In JavaScript push addes an element in the end of array, it is fast and easy, it does not require shift or move around existing elements

On the other hand the doublerInsert use unshift method. In JavaScript unshift addes the element at the beginning of the array, it requires a shift operation on all existing element, all existing elements need shift one position to the right to make space for the new element

That is why the doublerInsert funciton is slower