

Time Complexity Measurement: Execution Time

	tinyArray	smallArray	mediumArray	largeArray	extraLargeArray
doublerAppend	6.9 μ s	8.8 μ s	60 μ s	499.5 μ s	4.5988 ms
doublerInsert	4.2 μ s	15.9 μ s	166.7 μ s	11.0756 ms	1.3310043 s

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 doublerAppend function using .push() scales better than doublerInsert using the .unshift() method. You can tell by the time gap when you compare how each function is performing relative to its last entries as well as how it's performing relative to the other function. While doublerAppend also increases as we test larger arrays, the increment that it increases by is much smaller relative to doublerInsert. Each function handles tiny and smallArray with a pretty negligible difference, doublerInsert even a bit faster for tinyArray, but once we move on to mediumArray the difference becomes more apparent.

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

The .unshift() method is so much slower because every time it adds a new item to an array at the beginning, array[0], all other array items are moved over by one, changing each of their index values. In the .push() method, the new item is added to the end of the array and none of the other index values are changed: the method finds the end of the array and then adds the new item.