

Notes

extraLargeArray: the insert time takes much longer than the append time.

Time	tinyArray	smallArray	mediumArray	largeArray	extraLargeArray
Insert	36.6 μ s	47.7 μ s	170.5 μ s	8.8078 ms	1.002455 s
Append	86.2 μ s	96.1 μ s	131.4 μ s	726.1 μ s	3.2372 ms

As the numbers increased the both of the functions runtime increased but **doublerInsert** was taking longer than the **doublerAppend**. When it comes to the pattern I believe that the **doublerInsert** function was performing with the time complexity of Quadratic $O(n^2)$ because of how the runtime increased with the number increase. The **doublerAppend** seemed to be running as a Linear $O(n)$ because of how quickly it was able to run the function even with the substantial number increase. With that analysis I believe that $O(n)$ scales better because if there are any increases it will be more efficient with the amount of time it takes to complete the task.