Runtime Complexity of *push* vs. *unshift*

Runtime Results	**insert *f***	**append *f***
`extraLargeArray`	757.0312 ms	2.3406 ms
`largeArray`	7.1789 ms	452 μs
`mediumArray`	127.1 μs	123.8 μs
`smallArray`	34.3 μs	82 μs
`tinyArray`	28.1 μs	70.5 μs

The runtime test demonstrates that the *append* function scales much better than *insert*. That is because the *push* method only inserts an element at the end of the array, versus the *unshift* method which also has to increment all the existing elements in the array.

The runtime complexity of **push()** is constant — **O(1)** — whereas the runtime complexity of **unshift()** is linear — **O(n)** — taking longer to run the bigger the array size gets.

The loop makes the time complexity of the *insert* function O(n^2) and the time complexity of the *append* function O(n).