Arrays	Insert(.unshift)	Append(.push)
tinyArray	insert 38.667 μs	append 76.416 μs
smallArray	insert 46 μs	append 91.25 μs
mediumArray	insert 216.875 μs	append 153.458 μs
largeArray	insert 10.387375 ms	append 678 μs
extraLargeArray	insert 1.082815083 s	append 13.659916 ms

This was an interesting observation that definitely shows a great example of time complexity - and the importance of it. Once the input (array) reached over 999, the time for .unshift() to process began to take much longer than .push().

Extra Credit:

.unshift() is an O(1) time complexity, therefore it was faster as long as the constant (n) was < 1,000. However, .push() is an O(n) time complexity, which makes it much faster when the constants (n) are > 1,000. That's because O(1) is a constant time complexity - meaning that as input size increases, its time to process stays the same (constant). O(n), on the other hand, is a linear time complexity - meaning that as the input size increases, the time to process also increases. So .push() maintained the same time complexity no matter how big the input was whereas .unshift() took much longer after the input surpassed 999.