array	insert results	append results
tinyArray	5 microseconds	6.8 microseconds
smallArray	8.8 microseconds	4.7 microseconds
mediumArray	111.6 microseconds	31.4 microseconds
largeArray	8.1968 milliseconds	121.7 microseconds
extraLargeArray	947.7061 milliseconds	3.0692 milliseconds

Results for the tinyArray
insert 5 µs
append 6.8 µs

Results for the smallArray
insert 8.8 µs
append 4.7 µs

Results for the mediumArray
insert 111.6 µs
append 31.4 µs

Results for the largeArray
insert 8.1968 ms
append 121.7 µs

Results for the extraLargeArray
insert 947.7061 ms
append 3.0692 ms

Summary >

With the tiny array both functions have similarly low execution times. As the size of the input increases the execution time for the doublerInsert function shows significantly larger increases in time compared to the doublerAppend function. The doublerAppend function seems to scale better and more evenly than the doublerInsert function based on the time differences shown in the table above.

Extra credit + further summary >

The push method has a time complexity of o(1) because it adds an element to the end of an array making the execution time of a function that uses this remain consistent as the array size increases.
The unshift method inserts elements at the beginning of an array which makes each of the elements in the array shift over to make room. It has a time complexity of O(n) making the difference in execution time more pronounced as the array increases as more elements are shifting.