

Array Used	doublerAppend(nums) [.push]	doublerInsert(nums)[.unshift]
tinyArray	102.743 $\mu$ s	81.443 $\mu$ s
smallArray	129.855 $\mu$ s	85.884 $\mu$ s
mediumArray	174.793 $\mu$ s	247.301 $\mu$ s
largeArray	750.362 $\mu$ s	7.912166 ms
extraLargeArray	115.578014 ms	972.608452 ms

The doublerAppend function has a time complexity of  $O(n)$ . `.push()` has a time complexity of  $O(1)$  meaning it has constant time complexity. This means that the runtime depends on which array it's looping over.

The doublerInsert function has a time complexity of  $O(n^2)$  due to the `unshift()` method. This is because the `unshift` method will move the array after each loop. The runtime will be worse than the append function because `unshift` will need to iterate through the array for each loop.