	resultsAppend	resultsInsert
tinyArray	93.129 us	38.302 us
smallArray	108.845 us	49.609 us
mediumArray	150.12 us	159.419 us
largeArray	616.437 us	6.269 ms
extraLargeArray	9.158 ms	960.003 ms

The doublerAppend function starts out just a hair slower than the doublerInsert function, but as we move towards larger numbers it becomes apparent that doublerAppend is scaling linearly where as doublerInsert is scaling quadratically. So the doublerAppend function scales much better. This becomes apparent by the significant jump in values. Going from the 10,000 to 100,000 for doublerAppend resulted in roughly a jump that was 10x larger in the time it took to run it. This makes sense because the workload was indeed 10x larger. For doublerInsert though, the time it took was roughly 100x as long going from the largeArray to the extraLargeArray.

The reason why this is, is that with the unshift method has to move every single value to the space to the right. As the values get larger it takes longer and longer because there are more values that need to be shifted in order to accommodate the new value at the beginning of the array. With the push method on the other hand, it only needs to create one available space at the end of the array and fill it with the new results. This leaves the rest of the array untouched and the size does not affect the time of adding the additional values.