doubleAppend vs doubleInsert - with extraLargeArray

The run time for the doubleAppend function took about 1.8 milliseconds with the extraLargeArray passed in. The run time for the doubleInsert function took about 772.3 milliseconds with the extraLargeArray passed in. It took the doubleInsert function almost 700 times longer than the doubleAppend function to finish.

	doubleAppend	**doubleInsert**
tinyArray	53.1 microseconds	18.3 microseconds
smallArray	55.9 microseconds	23.9 microseconds
mediumArray	106.5 microseconds	166.7 microseconds
largeArray	639.7 microseconds	8.4 milliseconds
extraLargeArray	1.8 milliseconds	772.3 milliseconds

With the smaller arrays, the runtime for the doubleInsert functions is about twice as fast as the doubleAppend function. But as the arrays passed in get larger, the runtime for the doubleInsert function increases much faster than the doubleAppend function. At a small scale it might be beneficial to use the doubleInsert function, but at a large scale, doubleAppends runtime doesn't increase as fast which makes it significantly faster at a large scale.

The reason why doubleInsert is much slower at a large scale, is because in order to use ".unshift" and insert an element at the beginning of an array, the function must first move every element already existing in the array one index forward, one by one, so that there can be any empty spot at the beginning of the array for the new element. So as the array gets larger, the function has to go through more and more elements before inserting a new element. Whereas the doubleAppend function, which uses ".push", can simply open a new spot at the end of the array without having to go through all the other elements already existing in the array.