Jackie Bearnson 5/27/22 F22-Assessment-CS

Timing results:

3.

	tinyArray	smallArray	mediumArray	largeArray	extraLargeArray
doublerInsert	63.9 µs	76.6 µs	265.4 µs	13.6015 ms	2.657954 s
doublerAppend	237.5 µs	232.1 µs	201.3 µs	750.7 µs	4.5065 ms

Analysis:

4. doublerInsert starts smaller than doublerAppend, but it runs progressively more slowly as the size of the array increases resembling a function of the input size squared. doublerAppend seems to increase in time much more linearly as the size of the input array increases. This leads me to the conclusion that doublerInsert is likely O(n) time scaling whereas doublerAppend is most likely O(1).

5. Doubler Insert is O(n), and much slower as the input size increases because it uses the unshift method. Unshift works by inserting a value at the beginning of the array, which requires the code to then go through all other values in the array and increment the index of each by one. Because it has to loop through the entire array each time unshift is run, it creates an O(n) time scaling.