Rafael Daminelli – Assessment 7 – Part 1

Results for the extraLargeArray:

insert 1.2066682 s

append 3.1979 ms

The time to run the doublerInsert function, passing in the extraLargerArray, is much longer than to run the doublerAppend function passing in the same array, approximately 377 times longer, 1206.7 ms / 3.2 ms = 377.1, and over 1.2 second in absolute terms.

Table with runtimes for the 2 functions passing in different array sizes:

	Runtime for the doublerInsert function in µs	Runtime for the doublerAppend function in µs
tinyArray	37.3	96.7
smallArray	50.2	109.4
mediumArray	215	177.8
largeArray	10,729.2	571.9
extraLargeArray	1,229,633	3,483.3

Table pattern explained:

The results show that when the arrays passed in are relatively small, the doublerInsert function runs faster, but for larger arrays the doublerAppend function runs much faster. The difference is especially critical when considering larger arrays, as they require more noticeable runtimes. The doublerInsert function runtimes increases faster with larger arrays, this option makes the code less efficient. Thus, the doublerAppend function scales better in this case.

Extra Credit:

The doublerInsert uses the unshift method while the doublerAppend uses the push method. The unshift method makes the function much slower because it has to move all items one index over. This makes the doublerInsert much slower especially for very large arrays. The push method on the other hand, is much faster because it is just appending an item at the end without having to move the items already in the array.