

	doublerAppend runtimes	doublerInsert runtimes
tinyArray	149.7 μ s	65.8 μ s
smallArray	197.1 μ s	86.5 μ s
mediumArray	242.1 μ s	304.9 μ s
largeArray	986.2 μ s	17.7762 ms
extraLargeArray	6.2322 ms	1.8021004 s

doublerAppend, the function that uses the push method to add to an array, scales about linearly with an average runtime factor of $0.0627(n)$, where n is the number of items in the array passed in, and the factor is in microseconds. doublerInsert, the function that uses the unshift method to add to an array, scales about quadratically with microsecond factors of $0.0002(n^2)$ and $0.0253(n)$, where n is the number of items in the array. Thus, the doublerAppend function scales better because as the number of items in the array increases, the runtime increases steadily. Conversely, the doublerInsert runtime increases exponentially as the number of items in the array increases. This is most easily seen by comparing the runtime of the two functions when the extraLargeArray is used: the doublerInsert function has a runtime that is almost 290 times greater than that of the doublerAppend function. The slow runtime of doublerInsert could be due to the fact that every time an item is added to the front of an array using the unshift method, the indices of all the other array items need to be changed, and that takes time. When the push method is used, the only index that needs to be changed is that of the last item that is added in.