Step 1

Runtime Analysis:

Results Table:

|  |  |  |
| --- | --- | --- |
| **Array** | **Insert (Unshift)** | **Append (Push)** |
| tinyArray | 3.893 µs | 4.304 µs |
| smallArray | 9.819 µs | 7.37 µs |
| mediumArray | 110.604 µs | 83.029 µs |
| largeArray | 6.86783 µs | 626.348 ms |
| extraLargeArray | 834.753536 ms | 3.280125 ms |

Analysis:

Noted patterns: tiny array differences from insert to push are very minimal. Small array has noted reduced append function time. Medium Array has noted difference from insert to Append functions. Large & extra large arrays have significant noted difference from insert to append functions. The larger the array, the greater the difference in function scalability. Medium Array appears to scale best on both insert and append functions.

Extra Credit:

The Insert function is using unshift which adds the element at the start of the array. Append is using push which adds the element at the end of the array. Unshift will have a slower time because it also has to move all the elements after the new element is added. Push would scale much better in the long run because it doesn’t require moving the entire array elements but rather adds a new element on the end.