**RUNTIME REPORT – WEEK 7 ASSESSMENT:**

|  |  |  |
| --- | --- | --- |
| **ARRAY** | **INSERT** | **APPEND** |
| **extraLargeArray** | 847.0927 ms | 3.2585 ms |
| **largeArray** | 8.6937 ms | 649.1 μs |
| **mediumArray** | 183.9 μs | 156.3 μs |
| **smallArray** | 46.6 μs | 109.1 μs |
| **tinyArray** | 33.4 μs | 88.6 μs |

**OBSERVATIONS:**

Insert has a runtime of O(n), append has a runtime of O(1).

The pattern for both “insert” and “append” **is the larger the array, the longer the function takes to run**. This makes sense. However, the “insert” function takes much longer to run than the “append” function after they get past the ‘tiny’ and ‘small’ arrays. So even though *at first* it appears the “insert” function is faster (when starting with small arrays), **the “append” function scales much better.**

**EXTRA CREDIT:**

The slower function is so slow because to insert the value, it has to change the index of all the other values in the array. So, more is happening than just adding the value to the end like the append function.