Assessment Run Time

1. Tiny Array: insert = 28.7 us

append = 75.5 us

1. Small Array: insert = 35.8 us

append = 80.4 us

1. Medium Array: insert = 146.5 us

append = 116.6 us

1. Large Array: insert = 6.7849 ms

append = 503.6 us

1. XL Array: insert = 601.1407 ms

append = 3.2907 ms

as the array gets larger the runtime for both insert and append increases. When we start with the smaller arrays the insert function is a little faster but once we get up to the medium array the append method starts doing better but the difference between them is pretty insignificant at this size array. However as we get up to the larger arrays the append method scales up better with the insert method getting a big spike in runtime from this point on, and the difference between the two keeps growing.

Because of this the append method scales a lot better. We aren’t concerned with the runtime on small scales so we can see why it doesn’t matter that the insert method seems to do well in the beginning.

The reason the append scales better is because it is using the push method, so it is simply adding the new value to the end of the array allowing for all the previous indexes to stay the same, while the insert method is using unshift to put values on the begging of the array. So it has to re index every value after the new one and as you can see the more values it has to re index the slower it will run