

|            | Extra Large Array<br>5.100498084 s | Large Array<br>7.084458 ms | medium<br>130.625 µs | small<br>10.542 µs | tiny<br>5.167 µs |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|------------|------------------------------------|----------------------------|----------------------|--------------------|------------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| Insert     |                                    |                            |                      |                    |                  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| append     | 2.009708 ms                        | 457.208 µs                 | 145.167 µs           | 54.542 µs          | 46.917 µs        |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Conclusion |                                    |                            |                      |                    |                  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Based on the results above the push method is faster. After doing more research when adding to the end of the array it will rarely need to re-allocate memory copy over. While unshift in pushing elements to the front of the array it will always need to reallocate memory and copy data over because there's already an element at N position.