|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Array Size | Tiny Array | Small Array | Medium Array | Large Array | XL Array |
| Insert | insert 28.208 μs | insert 43.792 μs | insert 193.416 μs | insert 10.345708 ms | insert 1.262274916 s |
| Append | append 70.5 μs | append 80.209 μs | append 128.958 μs | append 707.167 μs | append 2.583042 ms |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

1000 μs = 1 ms

1000 ms = 1 s

As the size of the array grows from tiny to extra-large, the runtime of both the insert and append functions increase linearly. Because the runtime scales proportionally with the size of the array, we can classify the function as O(n). When the array is shortest, the insert function has a more efficient runtime because there are fewer elements to shift. When the array is at its longest length, the append is the more efficient runtime because it does not modify the list and instead adds the new entry to the end of the list.