Runtimes
Values rounded to the 3rd decimal

- Extra large array
    - insert: 1.435 ms, 1.502 ms, 1.466 ms
    - append: 4.274 ms, 4.902 ms, 4.928 ms
- Large array
    - insert: 12.865 ms, 12.861 ms, 12.613 ms
    - append: 745.5 µs, 780.4 µs, 716.4 µs
- Medium array
    - insert: 275.7 µs, 221.3 µs, 224.9 µs
    - append: 203.5 µs, 182.9 µs, 184.4 µs
- Small array
    - insert: 59.9 µs, 73.6 µs, 57.9 µs
    - append: 135.3 µs, 252.9 µs, 132.9 µs
- Tiny array
    - insert: 43.8 µs, 54.2 µs, 44.7 µs
    - append: 113.4 µs, 229.1 µs, 107.7 µs

It seems that generally speaking, insert is more effective and runs faster for a majority of the array sizes. However, in the medium array, append has a slight advantage, and a huge advantage for the large array. The smaller arrays are all relatively consistent, and the pattern returns with the extra large array. Despite the outlier of the large array and, to a lesser degree, the medium array, the insert function is more efficient and running through the arrays.