extraLargeArray
Insert 1.006689167
Append 2.174292

largeArray
insert 7.979167 ms
append 551.5 μs

Medium
insert 168.667 μs
append 145.667 μs

Small
insert 27.375 μs
append 61.5 μs

tiny
insert 20.875 μs
append 54.916 μs

	extraLargeArray	largeArray	Medium	Small	tiny
Insert	1.006689167	7.979167 ms	168.667 μs	27.375 μs	20.875 μs
Append	2.174292ms	551.5 μs	145.667 μs	61.5 μs	54.916 μs

The function Append scales better because it's an O(n) type, which is better for dealing with really really big numbers.However, Insert has a scalability of O(n^2), therefore slower in the long run. We can see this behavior in the extra large array section where the append has constant results at tiny, small, medium, and large. Insert has a constant growth in time. Insert uses unshift, which shifts ALL numbers of the array a step forward to make space for the one number that needs to jump in. compared to the push function that only adds the number at the very end of the array.