

Results for the tinyArray

insert 50.918 μ s	faster
append 125.782 μ s	slower

Results for the smallArray

insert 131.657 μ s	faster
append 157.196 μ s	slower

Results for the mediumArray

insert 244.899 μ s	slower
append 237.767 μ s	faster

Results for the largeArray

insert 8.979442 ms	slower
append 899.687 μ s	faster

extraLargeArray results :

Results for the extraLargeArray

insert 1.539624875 s	slower
append 36.259314 ms -> 0.036 s	faster

While the input of each function is relatively small it is more advantageous to use the `.insert` method for arrays, but as we scale up the size of the arrays the runtime starts shifting in favor of using the `.append()` method instead of `.insert()`. You can tell `append` scales better by the runtimes we see in the above example, the larger the array the more advantageous to use `.append`. The `insert` method results in a slower time as the arrays increase because the computer has to shift every single value in the array to the right and change the memory of the value's index position when it inserts a new value to the very front of the array. `.append` simply adds the new value to the end of the array which only requires recording one new index value.