CS Assessment

tinyAray
- Insert: 103 µs
- Append: 105 µs

smallArray
- Insert: 116 µs
- Append: 116 µs

Medium
- Insert: 215 µs
- Append: 169 µs

Large
- Insert: 6 ms
- Append: 601 µs

ExtraLarge
- Insert: 919 ms
- Append: 6 ms

The pattern I see is that the higher the number gets, the slower both functions get, obviously. However, the doublerInsert function doesn't scale nearly as well as the doublerAppend function. This is because the doublerInsert function is shifting every index over one as opposed to the doublerAppend function which is just adding a number to the end of the array. The larger the number, the more 'shifts' need to occur in the doublerInsert function, thus, causing scaling issues.