Function	Extra-large	Large	Medium	Small	Tiny
Append	2.212917 ms	411.25 µs	108.167 µs	77.042 µs	62.333 µs
Insert	767.516208 ms	6.41225 ms	165.875 µs	29 µs	21.75 µs

The execution time of the **doublerAppend** function increases gradually as the array size grows and scales relatively linearly with the array size. The increase in time follows a predictable pattern.
The execution time of the **doublerInsert** function increases significantly as the array size grows, indicating worse than linear scaling. The time it takes is not as predictable.
Therefore, the **doublerAppend** function scales better and is more efficient than **doublerInsert** for larger arrays.

The **doublerInsert** function is slower because it inserts elements at the beginning of the new array. It requires shifting all existing elements to the right for each insertion, which becomes increasingly time-consuming as the array size grows.
On the other hand, the **doublerAppend** function appends elements to the end of the new array. This approach is more efficient because it doesn't involve shifting any elements. The execution time for **doublerAppend** remains more consistent and predictable, regardless of the array size.