

After examining the results from running the code on arrays of different sizes, a clear pattern emerges regarding the scalability of the two functions, `doublerAppend` and `doublerInsert`. As the size of the input arrays increases, both functions take longer to execute, which is to be expected given their linear time complexity. However, there is a distinct difference in how the functions scale. The `doublerAppend` function exhibits a more consistent and manageable increase in execution time as the array size grows. On the other hand, the `doublerInsert` function's execution time increases significantly faster with larger array sizes. This indicates that the `doublerInsert` function scales less efficiently compared to `doublerAppend`. The noticeable disparity becomes more evident as the array size reaches the "extraLargeArray" size. Overall, the performance comparison suggests that `doublerAppend` is the more scalable option of the two functions, as it maintains a relatively steady increase in execution time with increasing array size, making it the preferred choice for larger arrays where efficiency is crucial.