Results for the extraLargeArray

insert 870.574751 ms

append 5.324042 ms

Based on the timing results, we can see how each function scales with input size. How it can be seen, the doublerAppend function works reliably shows much shorter execution times in comparison with the doubleInsert that work as the estimate of the input cluster increments. For illustration, for the extraLargeArray, doublerAppend takes almost 5.32 milliseconds while for doubleInsert takes longer,around 870.57 milliseconds. This design holds genuine for other cluster sizes as well, with doublerInsert reliably taking essentially more time to execute than doublerAppend.   
  
The reason for this distinction in adaptability is related to the operations each work performs. doublerAppend includes components to the conclusion of the cluster utilizing the thrust strategy, which is a proficient operation that doesn't require moving existing components. In differentiate, doublerInsert employments the unshift strategy, which embeds components at the starting of the cluster, causing all existing components to be moved to create room for the modern one. As the cluster estimate increments, this operation gets to be progressively time-consuming, driving to the watched execution corruption.   
  
In outline, doublerAppend scales much superior than doublerInsert, making it the more proficient choice, particularly for expansive clusters. The comes about clearly illustrate that the choice of cluster control strategies can have a critical affect on the adaptability and execution of code.