Performance Complexity Discussion

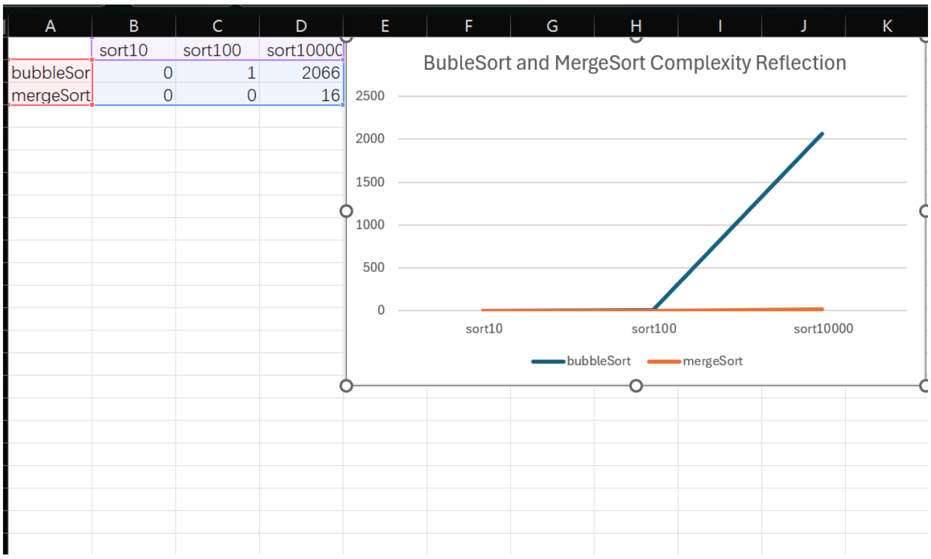
Bubble Sort and Merge Sort Complexity Reflection

When sorting large datasets, the performance of the algorithm becomes crucial. By comparing the performance of Bubble Sort and Merge Sort across different data set sizes, we can gain insights into the time complexity characteristics of these two sorting algorithms.

Bubble Sort is a simple sorting algorithm that works by repeatedly stepping through the list to be sorted, comparing adjacent elements, and swapping them if they are in the wrong order. The average and worst-case time complexity of Bubble Sort is O(n^2), where n is the number of elements in the list. In this experiment, as the size of the dataset increased, the performance of Bubble Sort deteriorated significantly, especially when handling 1000 elements, where its execution time far exceeded that of Merge Sort.

On the other hand, Merge Sort employs a divide-and-conquer strategy, dividing the list into smaller sublists that are each sorted independently, and then merging these sorted sublists back together. Merge Sort has a time complexity of O(n log n) in all cases, making it more efficient for handling large datasets. From the experimental results, it is evident that Merge Sort maintained a consistently lower execution time across all data sizes, demonstrating better stability and scalability.

In summary, while Bubble Sort may perform adequately on small datasets, its performance disadvantages become apparent with larger data sizes. In contrast, Merge Sort, due to its lower time complexity, exhibits superior performance on larger datasets. Therefore, when selecting a sorting algorithm for practical applications, it is essential to consider the dataset size and performance requirements.



	sort10	sort100	sort1000C
bubbleSor	0	1	2066
mergeSort	0	0	16