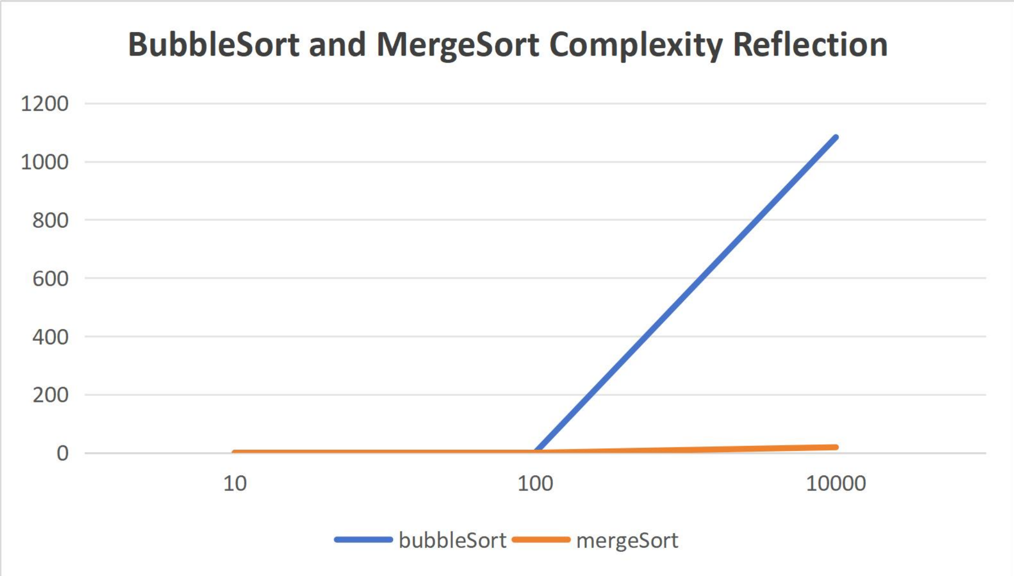
BubbleSort and MergeSort Complexity Reflection



In our experiment, we analyzed the performance of BubbleSort and MergeSort using three datasets: `sort10.txt`, `sort1000.txt`, and `sort10000.txt`. BubbleSort, with its $O(n^2)$ complexity, demonstrated predictable results. As expected, it performed reasonably with the smallest dataset but showed significant slowdowns as the size increased. This inefficiency is especially visible with `sort10000.txt`, where the number of required operations increased dramatically. Such behavior aligns with our understanding that BubbleSort is less suitable for large datasets due to its quadratic nature.

In contrast, MergeSort exhibited much more consistent and scalable performance across all datasets, thanks to its $O(n \log n)$ complexity. This divide-and-conquer algorithm efficiently manages larger datasets by recursively breaking down the data into smaller parts, sorting them independently, and then merging them back together. The performance gap between BubbleSort and MergeSort becomes more pronounced as the dataset size grows, highlighting MergeSort's superiority for handling larger volumes of data efficiently.

The results corroborate theoretical expectations and emphasize the importance of choosing appropriate algorithms based on dataset sizes. While BubbleSort might be used for educational purposes or very small datasets, MergeSort is clearly more adaptable and efficient for practical applications involving significant amounts of data. This reflection underscores the relevance of understanding algorithmic complexity when approaching real-world problems.