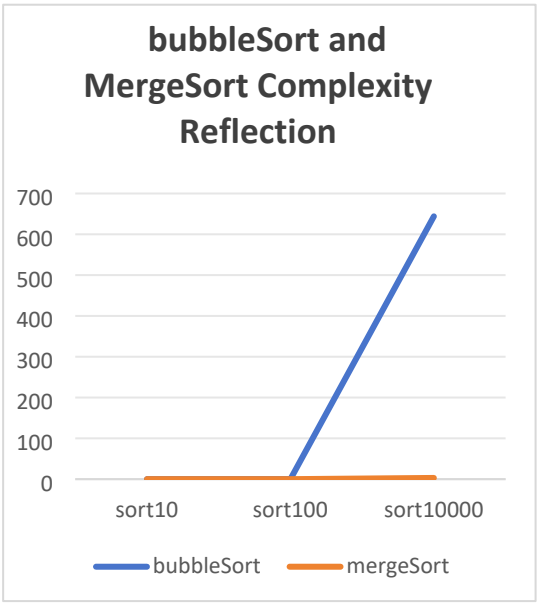
When dealing with the data sorting problem, choosing the appropriate algorithm is crucial, as it directly affects the efficiency and performance of the program. This PDF will include a complexity discussion regarding the performance of two sort functions. Then, the document will reflect on BubbleSort and MergeSort Complexity with an image plot and related written text discussing results of the *sortComparison* for the files *sort10.txt*, *sort100.txt*, and *sort10000.txt*. Finally, the document will discuss the advangtages of merging sorting.

The performance of ranking algorithms is usually measured by time complexity and space complexity. Time complexity describes the relationship between the time required by the algorithm execution and the amount of input data, while space complexity describes the relationship between the storage space occupied during the execution of the algorithm and the amount of input data.In this comparison, Bububbling and merging sorting were used to sort an array of 10,100,10,000 elements, and Then the processing time is made into the following table and chart for comparison.

	sort10	sort100	sort10000
bubbleSor	0	0	644
mergeSort	0	0	3

For small-scale datasets, such as file sort10.txt with 10 elements, simple sorting algorithms such as bubble sorting or insertion sorting may perform quite efficient given their small constant factor and simplicity to implement. However, as the amount of data increases, such as when processing the file sort100.txt containing 100 elements, the performance of these simple algorithms will gradually decline, because their time complexity is O (n ^ 2), which will lead to significant performance bottleneck in the amount of data, but because the amount of data is still small, the processing time has minor difference. For large-scale datasets, such as file sort10000.txt with 10,000 elements, more efficient sorting algorithms such as merging sorting are more efficient. These algorithms usually have better average time complexity, with a time complexity of O (n log n), but a higher space complexity, so it requires additional storage space to merge the sorted subarrays.



By comparing the sorting results of different files and then plotting the following chart, it can be observed that the performance of the bubble sorting algorithm decreases significantly with the increase of data volume, while the merging sorting can maintain excellent performance.

This suggests that selecting ranking algorithms with low time complexity is crucial when processing large-scale datasets. At the same time, the space complexity of the algorithm cannot be ignored. Therefore, in practical application, the most appropriate ranking algorithm should be selected according to the comprehensive consideration of data scale and system resources. Therefore, merging sorting might be more efficient than bubble sorting when processing large-scale datasets.However, when dealing with small-scale data, both of them are highly efficient.