SortComparison

Analysis of Sorting Algorithm Complexities

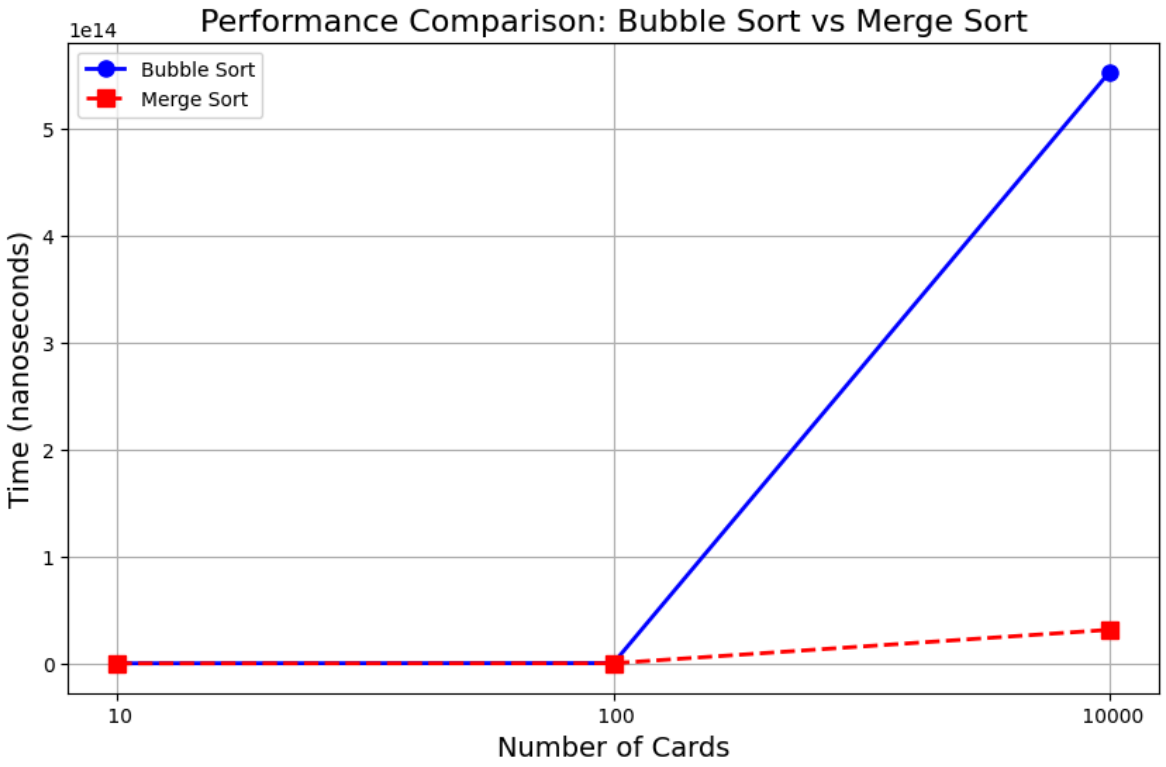
In the realm of sorting algorithms, Bubble Sort and Merge Sort stand out as two fundamental and widely studied methods. Bubble Sort is an elementary and intuitive sorting technique that operates by repeatedly stepping through the list, comparing adjacent elements, and swapping them if they are in the wrong order. This process continues until the list is sorted. The time complexity of Bubble Sort is $O(n^2)$, making it inefficient for large datasets. Despite its simplicity, Bubble Sort is often used for educational purposes and in scenarios where the dataset is relatively small or nearly sorted, and the overhead of more complex algorithms is unjustified.

In contrast, Merge Sort employs a divide-and-conquer strategy to efficiently sort large datasets. It recursively divides the list into smaller sublists until each sublist contains a single element. These sublists are then merged in a manner that results in a sorted sequence. Merge Sort boasts a time complexity of $O(n \log n)$ in the best, average, and worst cases, ensuring consistent performance regardless of the input size. Its stable sorting nature and efficient handling of large volumes of data make Merge Sort a preferred choice in applications where performance and reliability are critical.

Experimental sorting of datasets of varying sizes, accompanied by charts created in Excel, illustrates the performance differences between Bubble Sort and Merge Sort. As the input size increases, the sorting time for Bubble Sort grows quadratically, whereas Merge Sort exhibits a growth rate proportional to $n \log n$. The charts clearly demonstrate that Merge Sort outperforms Bubble Sort significantly as the dataset scales, highlighting its superior efficiency for large-scale sorting tasks. These visual representations validate the theoretical time complexities and underscore the importance of selecting appropriate algorithms based on the specific requirements and data characteristics of the application at hand.

Performance Analysis and Graphs

Filename	Bubble Sort Time (ns)	Merge Sort Time (ns)
sort10.txt	11500	7100
sort100.txt	842300	137400
sort10000.txt	7899715600	34340600


Performance Comparison: Bubble Sort vs Merge Sort
1e14
Bubble Sort
Merge Sort
Time (nanoseconds)
0
1
2
3
4
5
10
100
10000
Number of Cards