Sorting Algorithms Performance Report

This experiment compares the performance of **Bubble Sort** and **Quick Sort**, two fundamental sorting algorithms, across different dataset sizes.

Algorithm Functionality:

Bubble Sort is a simple, comparison-based algorithm that repeatedly steps through a list, compares adjacent elements, and swaps them if they are in the wrong order. The process continues until the entire list is sorted. Quick Sort, in contrast, is a divide-and-conquer algorithm. It selects a pivot element, partitions the list into elements smaller or larger than the pivot, and recursively sorts the partitions. The partitions are then combined to form a fully sorted list. Both algorithms produce identical sorted outputs but differ greatly in efficiency.

Efficiency Analysis:

The timing tests reveal that Bubble Sort has a quadratic time complexity of $\mathbf{O(n^2)}$. While it handles small datasets (100–300 elements) reasonably quickly, its runtime increases sharply as the input size grows, making it impractical for larger datasets. Quick Sort, with an average time complexity of **O(n log n)**, handles both small and large datasets efficiently. Its execution time increases much more slowly with input size, demonstrating excellent scalability.

Comparison and Real-World Applications: Bubble Sort is primarily used in educational settings to illustrate basic algorithmic concepts due to its simplicity. Quick Sort, however, is widely used in real-world applications such as standard library sorting functions, database indexing, and scenarios requiring efficient sorting of large datasets.

The experiment clearly shows that Quick Sort outperforms Bubble Sort as dataset size grows, confirming its practical utility.

Conclusion:

The performance graph illustrates that Bubble Sort's runtime grows steeply with input size, while Quick Sort maintains efficient performance even for larger arrays. In summary, Bubble Sort is suitable only for small datasets or teaching purposes, whereas Quick Sort is the preferred choice for practical, large-scale sorting tasks.