Merge Sort

Merge Sort is a divide and conquer algorithm that divides the input array into two halves then recursively sorts each half, and then merges the sorted halves to produce the final sorted array.

Key Characteristics of Merge Sort:

- Time Complexity: O(n log n) in all cases
- Space Complexity: O(n)
- Suitable for large datasets and linked lists

Advantages of Merge Sort

- Guaranteed performance regardless of input order
- Stable sorting algorithm

Disadvantages of Merge Sort

- Requires extra memory
- Slower for small datasets compared to other algorithms

Quick Sort

Quick Sort is a divide-and-conquer algorithm that selects a pivot element and partitions the array into elements less than the pivot and greater than the pivot. The subarrays are then sorted recursively.

Key Characteristics of Quick Sort:

- Time Complexity: Best/Average O(n log n), Worst $O(n^2)$
- Space Complexity: O(log n) (recursive stack)
- Stable: No
- Very fast in practice

Advantages of Quick Sort

- Fastest sorting algorithm in most practical scenarios
- In-place sorting (low memory usage)

Disadvantages of Quick Sort

- Worst-case performance is $O(n^2)$
- Not stable

3. Comparison Table

Aspect	Merge Sort	Quick Sort
Time Complexity	O(n log n)	O(n log n) avg, O(n^2) worst
Space Complexity	O(n)	O(log n)
Stability	Stable	Not Stable
Performance	Consistent	Very fast in practice

4. Conclusion

Both Merge Sort and Quick Sort are efficient sorting algorithms with their own strengths. Merge Sort is preferred when stability and predictable performance are required, while Quick Sort is often chosen for its superior average-case speed and low memory usage.