**Understand Sorting Algorithms**

**Bubble Sort**

* Repeatedly compares adjacent elements and swaps them if out of order.
* **Time Complexity**:
  + Best: O(n) (already sorted)
  + Worst & Avg: O(n²)
* **Simple but inefficient** for large data.

**Insertion Sort**

* Builds sorted array one element at a time.
* **Time Complexity**:
  + Best: O(n), Worst: O(n²)
* **Good for small datasets**.

**Quick Sort**

* Divide and conquer; picks a pivot and partitions array.
* **Time Complexity**:
  + Best & Avg: O(n log n), Worst: O(n²)
* **Efficient and commonly used**.

**Merge Sort**

* Divide and conquer; always divides into halves, then merges.
* **Time Complexity**: O(n log n) in all cases.
* Uses extra space.