**Understand Sorting Algorithms**

**Sorting Algorithms Overview:**

1. **Bubble Sort:**
   * **Description:** A simple comparison-based algorithm that repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. The process is repeated until the list is sorted.
   * **Time Complexity:**
     + **Best Case:** O(n) (when the list is already sorted)
     + **Average Case:** O(n^2)
     + **Worst Case:** O(n^2)
2. **Insertion Sort:**
   * **Description:** Builds the final sorted array one item at a time. It takes each element and inserts it into its correct position in the already sorted part of the array.
   * **Time Complexity:**
     + **Best Case:** O(n) (when the list is already sorted)
     + **Average Case:** O(n^2)
     + **Worst Case:** O(n^2)
3. **Quick Sort:**
   * **Description:** A divide-and-conquer algorithm that selects a 'pivot' element and partitions the array into two sub-arrays, according to whether elements are less than or greater than the pivot. The sub-arrays are then sorted recursively.
   * **Time Complexity:**
     + **Best Case:** O(n log n)
     + **Average Case:** O(n log n)
     + **Worst Case:** O(n^2) (when the smallest or largest element is consistently chosen as the pivot)
4. **Merge Sort:**
   * **Description:** A divide-and-conquer algorithm that splits the array into halves, sorts each half, and then merges the sorted halves to produce the final sorted array.
   * **Time Complexity:**
     + **Best Case:** O(n log n)
     + **Average Case:** O(n log n)
     + **Worst Case:** O(n log n)