Exercise3

1. Understanding Sorting Algorithms

Sorting Algorithms Overview:

- **Bubble Sort:** A simple comparison-based sorting algorithm. It 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: O(n^2) in the worst and average cases, and O(n) in the best case (when the list is already sorted).
- **Insertion Sort:** Builds the final sorted list one item at a time. It picks elements from the unsorted part and inserts them into their correct position in the sorted part. Time complexity: O(n^2) in the worst and average cases, and O(n) in the best case.
- **Quick Sort:** A comparison-based sorting algorithm that uses divide-and-conquer. It selects a pivot element, partitions the array around the pivot, and then recursively applies the same strategy to the sub-arrays. Time complexity: O(n log n) on average and O(n^2) in the worst case (with poor pivot choices).
- **Merge Sort:** A comparison-based sorting algorithm that divides the array into two halves, recursively sorts each half, and then merges the sorted halves. Time complexity: O(n log n) in all cases.

4. Analysis

Time Complexity:

- **Bubble Sort:**
  - **Best Case:** O(n) (when the array is already sorted).
  - **Average Case:** O(n^2).
  - **Worst Case:** O(n^2) (when the array is sorted in reverse).
- **Quick Sort:**
  - **Best Case:** O(n log n) (when the pivot divides the array into two roughly equal halves).
  - **Average Case:** O(n log n).
  - **Worst Case:** O(n^2) (when the pivot is the smallest or largest element).

Why Quick Sort is Preferred:

- **Efficiency:** Quick Sort generally performs faster on average due to its O(n log n) time complexity, compared to Bubble Sort's O(n^2).
- **Divide-and-Conquer:** Quick Sort's approach of dividing the problem into smaller sub-problems and solving them recursively is more efficient for large datasets.
- **In-place Sorting:** Quick Sort sorts in place and doesn't require additional memory for an auxiliary array (unlike Merge Sort).