CS-1203 – Monsoon 2023 – Assignment 3

Kudakwashe Chakanyuka

kudakwashe.chakanyuka_ug25@ashoka.edu.in

October 18, 2023

Question 3

Merge Sort:

Best-case time complexity: O(n log n)

Average-case time complexity: O(n log n)

Worst-case time complexity: O(n log n)

Merge Sort is an efficient, stable sorting algorithm with a consistent time complexity. It divides the input array into two equal halves, sorts each half, and then merges the sorted halves. This process continues recursively until the entire array is sorted. The time complexity is O(n log n) because each level of recursion divides the array into two halves, and the merging step takes O(n) time.

Stability: Merge Sort is a stable sorting algorithm, which means it maintains the relative order of equal elements in the sorted array.

Quick Sort:

Best-case time complexity: O(n log n)

Average-case time complexity: O(n log n)

Worst-case time complexity: O(n^2)

Quick Sort is a fast and efficient sorting algorithm with good average-case performance. It works by selecting a pivot element, partitioning the array into two sub-arrays (elements less than the pivot and elements greater than the pivot), and recursively sorting these sub-arrays. The time complexity depends on the choice of pivot.

In the average and best cases, the pivot choice leads to well-balanced partitions and a time complexity of O(n log n). However, in the worst case, when the pivot choice consistently leads to unbalanced partitions, the time complexity can degrade to O(n^2).

Stability: Quick Sort is not stable, meaning it may change the relative order of equal elements in the sorted array.

Heap Sort:

Best-case time complexity: O(n log n)

Average-case time complexity: O(n log n)

Worst-case time complexity: O(n log n)

Heap Sort is an in-place sorting algorithm that relies on the properties of a binary heap data structure. It builds a max-heap (or min-heap), where the maximum (or minimum) element is at the root.

Repeatedly removing the root and restoring the heap property effectively sorts the array. The time complexity of Heap Sort is O(n log n) for all cases because the height of a binary heap is logarithmic in the number of elements.

Stability: Heap Sort is not stable, as it may change the relative order of equal elements in the sorted array.

In summary, Merge Sort and Heap Sort have consistent time complexities across different scenarios, making them reliable choices for sorting algorithms. Quick Sort is usually faster in practice due to its smaller constant factors, but it can exhibit its worst-case time complexity if the pivot choice is not well-balanced. The choice of sorting algorithm depends on the specific requirements and constraints of the task at hand.