CS-1203 – Monsoon 2023 – Assignment 3

Kudakwashe Chakanyuka

kudakwashe.chakanyuka_ug25@ashoka.edu.in

October 18, 2023

Question 2

- After implementing the insertion sort and comparing it with Bubble sort, both theoretical time complexity analysis and experimental data follows as such:

Insertion Sort:

Best-case time complexity: O(n)

- When the input array is already sorted, the algorithm only needs to compare and shift elements once for each element. It makes a single pass through the array and does no shifting.

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

- The algorithm needs to make approximately $\frac{n^2}{4}$ comparisons and $\frac{n^2}{4}$ swaps, where 'n' is the number of elements in the array. This results in quadratic time complexity. The actual number of operations may vary based on the initial order of elements.

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

- When the input array is in reverse order, the algorithm requires the maximum number of comparisons and shifts. It makes $\frac{n(n-1)}{2}$ comparisons and $\frac{n(n-1)}{2}$ swaps, leading to a quadratic time complexity.

Bubble Sort:

Best-case time complexity: O(n)

- When the input array is already sorted, Bubble Sort requires just one pass to confirm that the array is sorted. It checks adjacent elements and finds no swaps are needed, resulting in linear time complexity.

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

- Bubble Sort requires approximately $\frac{n^2}{2}$ comparisons and $\frac{n^2}{2}$ swaps. The number of comparisons and swaps grows quadratically with the size of the array, leading to quadratic time complexity.

Worst-case time complexity: $\mathrm{O}(n^2)$

- Bubble Sort requires the maximum number of comparisons and swaps when the input array is in reverse order. It makes $\frac{n(n-1)}{2}$ comparisons and $\frac{n(n-1}{2}$ swaps, resulting in a quadratic time complexity.

The best-case time complexity for both algorithms occurs when the input array is already sorted. In this case, only one pass is required to verify the array's sorted status, leading to a linear time complexity.