KJ's Educational Institutes

K J College Of Engineering & Management Research, Pune.

Department of E & TC

CLASS: S. E. (E &TC) SUBJECT:-DSA

Ex. No: 8 Date:

AIM

Sorting Algorithms

An online store wants to sort its product prices to help customers compare them easily. Choose suitable sorting techniques for small to medium datasets. Implement bubble sort, selection sort, and insertion sort to reorder product prices.

OBJECTIVES

- To compare and implement various sorting algorithms like **Bubble Sort**, **Selection Sort**, and **Insertion Sort** for sorting product prices in an online store.
- To understand how these algorithms work and their efficiency when applied to small to medium-sized datasets.

THEORY

Sorting is the process of arranging items in a particular order. In this experiment, the goal is to sort an array of product prices in ascending order. Three popular sorting techniques, each with distinct characteristics and time complexities, will be implemented:

1. **Bubble Sort**:
   - **Concept**: Repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. This is repeated until the list is sorted.
   - **Time Complexity**: Worst case and Average case: $O(n^2)$ Best case: O(n) (if the list is already sorted).
   - **Space Complexity**: O(1)(in-place sorting).
2. **Selection Sort**:
   - **Concept**: Divides the list into a sorted and an unsorted section. In each step, it selects the minimum element from the unsorted part and moves it to the sorted part.
   - **Time Complexity**: $O(n^2)$ (best, average, and worst case).
   - **Space Complexity**: O(1) (in-place sorting).
3. **Insertion Sort**:
   - **Concept**: Iterates through the list and builds a sorted portion by inserting each element into its correct position within the sorted section.
   - **Time Complexity**: Worst case and Average case: $O(n^2)$, Best case: O(n) (if the list is already sorted).
   - **Space Complexity**: O(1) (in-place sorting).

ALGORITHM

1. **Initialize the array**:

   Create an array of product prices.

2. **Implement each sorting algorithm**:

   Write separate functions for Bubble Sort, Selection Sort, and Insertion Sort to reorder the prices in ascending order.

3. **Compare the efficiency**:

   Observe the results of each sorting technique. While the time complexity is the same in the worst case for all three, the implementation will allow us to compare their behavior on small datasets.

INPUT:

- No. products
- Prize of each product

OUTPUT:

```
Enter the number of products: 5
Enter the prices of 5 products:
250 120 90 300 150

Bubble Sort:
Product Prices:
90.00 120.00 150.00 250.00 300.00

Enter the prices of 5 products:
250 120 90 300 150

Selection Sort:
Product Prices:
90.00 120.00 150.00 250.00 300.00

Enter the prices of 5 products:
250 120 90 300 150

Insertion Sort:
Product Prices:
90.00 120.00 150.00 250.00 300.00
```

CONCLUSION:-