Session - 4

Sorting

CONTENTS

1. Introduction to Sorting
2. Sorting Algorithms
3. Activity

01

Introduction to Sorting

Introduction to Sorting

- The process of arranging data in a specific order (ascending or descending).
- It helps in searching, organizing, and optimizing data processing.
- Common use cases:
  - Database indexing
  - Searching algorithms
  - Data analysis

02

Sorting Algorithms

Types of Sorting Algorithms

1. Bubble Sort
2. Selection Sort
3. Insertion Sort
4. Merge Sort
5. Quick Sort
6. Heap Sort

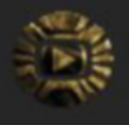
Bubble Sort (Basic Sorting Algorithm)

- Compare adjacent elements and swap if needed.
- Time Complexity:
  - Best: O(n) (already sorted)
  - Worst: $O(n^2)$
- Use Case: Small datasets, simple scenarios

Example:

Before: [5, 3, 8, 1]

After Pass 1: [3, 5, 1, 8]
After Pass 2: [3, 1, 5, 8]
After Pass 3: [1, 3, 5, 8]



Selection Sort

- Find the smallest element and swap it with the first element.
- Time Complexity: $O(n^2)$
- Use Case: Small datasets, less swapping required

Example:

Insertion Sort

- Pick an element and place it in the correct position in the sorted part.
- Time Complexity:
  - Best: O(n)
  - Worst: $O(n^2)$
- Use Case: Nearly sorted lists

Merge Sort (Divide and Conquer Approach)

- Split the array, sort recursively, and merge.
- Time Complexity: O(n log n)
- Use Case: Large datasets, stable sorting
- Example: Merge Sort

Quick Sort (Divide and Conquer Approach)

- Choose a pivot, partition the array, and recursively sort.
- Time Complexity:
  - Best: O(n log n)
  - Worst: $O(n^2)$ (if poor pivot selection)
- Use Case: Efficient for large datasets
- Example: Quick Sort

Choosing the Right Sorting Algorithm

Algorithm	Best case	Worst Case	Space complexity	stability	Use case
Bubble Sort	O(n)	$O(n^2)$	O(1)	Yes	Small data, easy implementation
Selection Sort	$O(n^2)$	$O(n^2)$	O(1)	No	Small data, fewer swaps
Insertion Sort	O(n)	$O(n^2)$	O(1)	Yes	Nearly sorted data
Merge Sort	O(n log n)	O(n log n)	O(n)	Yes	Large datasets
Quick Sort	O(n log n)	$O(n^2)$	O(log n)	No	General-purpose, efficient

03

Activity

Activity

</> Sort an Array

</> Minimum Absolute Difference

</> Next Greater Element

</> Wiggle Sort