Summary: Data Structures, Linked Lists, Stacks, and Queues

What is a Data Structure?

A **data structure** is a method of organizing and storing data in a computer so that it can be accessed and modified efficiently.
It determines how data is arranged in memory and what operations can be performed on it (such as insertion, deletion, searching, and traversal).

Types of Data Structures

1. Linear Data Structures

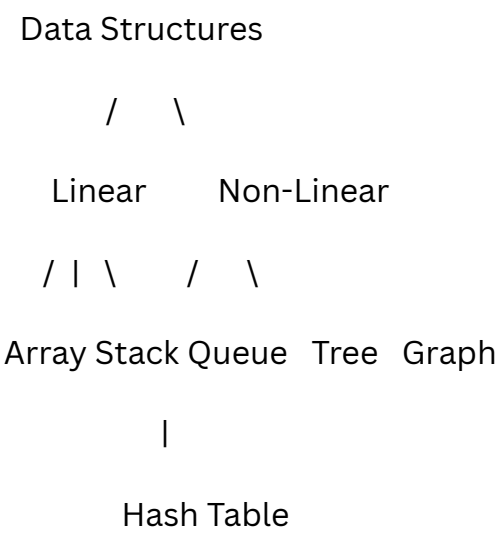
- Elements are arranged in a sequence.
- Examples:
  - **Array**
  - **Linked List**
  - **Stack**
  - **Queue**

2. Non-Linear Data Structures

- Elements are organized in a hierarchical or interconnected way.
- Examples:
  - **Tree**
  - **Graph**

3. Hash-Based Structures

- Store data using key-value pairs.
- Example:
- **Hash Table**



Structure	Description	Common Operations
Array	Fixed-size structure with indexed elements	Access, Update, Search
Linked List	Nodes connected with pointers	Insert, Delete, Traverse
Stack	LIFO structure (Last In, First Out)	Push, Pop, Peek
Queue	FIFO structure (First In, First Out)	Enqueue, Dequeue
Tree	Hierarchical structure	Insert, Traverse, Search
Graph	Nodes connected by edges	Search (DFS, BFS), Path
Hash Table	Key-value data storage	Insert, Delete, Lookup

Linked List

A **linked list** is a linear data structure where elements (nodes) are connected using pointers. Each node contains:

- Data
- A pointer to the next node (and possibly the previous node)

Types of Linked Lists

1. **Singly Linked List**
   - Each node points to the next node.
2. **Doubly Linked List**
   - Each node points to both the next and previous nodes.
3. **Circular Linked List**
   - The last node points back to the first node.

Applications

- Memory-efficient dynamic data structures
- Implementing stacks and queues
- Undo/Redo operations in editors
- Navigation systems (forward/backward)
- Process scheduling in operating systems

Queue

- Queue is a linear data structure using **First In, First Out (FIFO)** principle.
- Elements are inserted at the rear and removed from the front.

Common Operations:

- Enqueue: Insert at the rear.
- Dequeue: Remove from the front.
- Front: View the front element.
- Rear: View the last element.

Applications:

- Task and process scheduling
- Buffers in operating systems
- Printing queues and job management

Feature	Stack	Queue
Order	Last In, First Out (LIFO)	First In, First Out (FIFO)
Insert Operation	Push (top)	Enqueue (rear)
Delete Operation	Pop (top)	Dequeue (front)
Access Ends	One end only (top)	Both ends (front and rear)
Use Cases	Recursion, Undo, Expression	Scheduling, Buffers, Printers