6t**h** April 2025 (Sunday) Start Non-Linear DS from IIRS ISRO Dehradun, Golden Jubilee Hostel Reading Cum Library Room and completed Linear-DS

**🔹 1. Linear Data Structures**

Data is arranged in a linear sequence.

**a) Array**

* Fixed size.
* Elements stored in **contiguous memory**.
* Example:  
  int arr[5] = {1, 2, 3, 4, 5};

**b) Linked List**

* Dynamic size.
* Each element (node) contains **data + pointer** to the next node.
* Types: Singly, Doubly, Circular.

**c) Stack (LIFO: Last In, First Out)**

* Insert (push) and remove (pop) from the **top** only.
* Think of a pile of plates.

**d) Queue (FIFO: First In, First Out)**

* Insert from **rear**, remove from **front**.
* Variants: Circular Queue, Deque, Priority Queue.

**🔹 2. Non-Linear Data Structures**

Data is not stored sequentially.

**a) Tree**

* Hierarchical structure.
* **Root**, **Child**, **Leaf** nodes.
* Types: Binary Tree, Binary Search Tree (BST), AVL Tree, Heap.

**b) Graph**

* Collection of **nodes (vertices)** and **edges**.
* Can be:
  + **Directed / Undirected**
  + **Weighted / Unweighted**

**🔹 3. Hash-Based Data Structures**

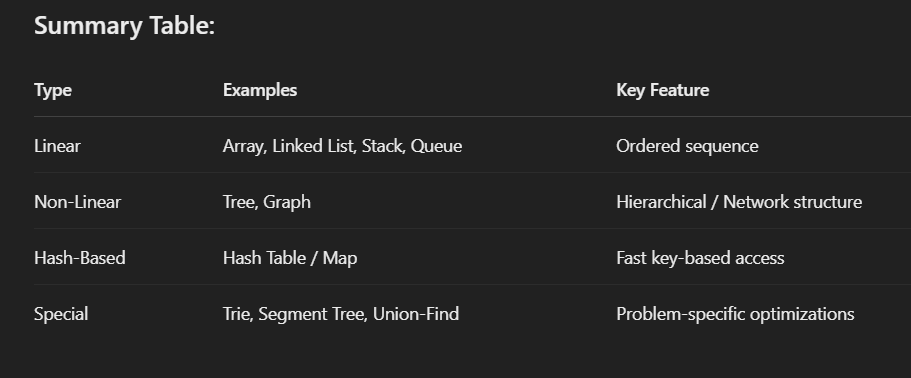
Used for fast lookups.

**a) Hash Table / Hash Map**

* Stores key-value pairs.
* Uses **hashing function** to index.
* Example: unordered\_map<int, string> mp;

**🔹 4. Advanced / Special Data Structures**

* **Trie** – For efficient retrieval of strings (like autocomplete).
* **Segment Tree / Fenwick Tree** – For range queries.
* **Disjoint Set (Union-Find)** – For dynamic connectivity problems.
* **Suffix Tree / Array** – For string pattern matching.



DS