Data Structure and Its Types

**Definition**: Data structure is a way of organizing, storing, and managing data efficiently to perform various operations such as insertion, deletion, searching, and sorting. It provides a systematic way of accessing and manipulating data.

**Types of Data Structures**:

1. **Arrays**:
   * Contiguous memory locations to store elements of the same data type.
   * Fixed size, elements accessed using an index.
   * Efficient for random access but less flexible for dynamic operations.
2. **Linked Lists**:
   * Elements (nodes) are stored non-contiguously with each node pointing to the next.
   * Dynamic size, efficient for insertion and deletion at any position.
   * Types: Singly Linked List, Doubly Linked List, Circular Linked List.
3. **Stacks**:
   * Linear data structure following Last In, First Out (LIFO) principle.
   * Operations: push (insert), pop (delete), peek (access top element).
   * Used in function calls, expression evaluation, and backtracking algorithms.
4. **Queues**:
   * Linear data structure following First In, First Out (FIFO) principle.
   * Operations: enqueue (insert), dequeue (delete), front (access front element).
   * Used in scheduling, resource allocation, and breadth-first search algorithms.
5. **Trees**:
   * Non-linear data structure composed of nodes connected by edges.
   * Hierarchical structure with a root node and child nodes.
   * Types: Binary Tree, Binary Search Tree, AVL Tree, B-tree, Red-Black Tree.
6. **Graphs**:
   * Non-linear data structure consisting of vertices and edges.
   * Represent connections between pairs of elements.
   * Types: Directed Graphs, Undirected Graphs, Weighted Graphs, Sparse Graphs.
7. **Hash Tables**:
   * Data structure that stores key-value pairs for efficient retrieval.
   * Uses a hash function to map keys to array indices (buckets).
   * Provides constant-time average-case complexity for insertion, deletion, and retrieval.