**Data Structures Study Material**

**Introduction**

Data structures are essential for organizing, managing, and storing data efficiently. They serve as the backbone of algorithm development, helping solve complex computational problems.

**1. Types of Data Structures**

**1.1. Primitive Data Structures**

* Integers, Float, Boolean, Character (Basic building blocks).

**1.2. Non-Primitive Data Structures**

* **Linear Structures**: Arrays, Linked Lists, Stacks, Queues.
* **Non-Linear Structures**: Trees, Graphs.

**2. Arrays**

* **Definition**: A collection of elements stored in contiguous memory locations.
* **Key Features**:
  + Fixed size.
  + Direct access via indices.
* **Advantages**:
  + Easy to implement.
  + Efficient for static data.
* **Applications**: Sorting, Searching (Binary Search).

**3. Linked Lists**

* **Definition**: A collection of nodes where each node contains data and a reference to the next node.
* **Types**:
  + Singly Linked List.
  + Doubly Linked List.
  + Circular Linked List.
* **Advantages**:
  + Dynamic size.
  + Efficient insertion/deletion.
* **Applications**: Implementation of stacks, queues, and graphs.

**4. Stacks and Queues**

**4.1. Stack**

* **Definition**: A collection based on LIFO (Last In, First Out) principle.
* **Applications**: Backtracking (Maze Solving), Expression Evaluation.

**4.2. Queue**

* **Definition**: A collection based on FIFO (First In, First Out) principle.
* **Applications**: Task Scheduling, Breadth-First Search (BFS).

**5. Trees**

* **Definition**: A non-linear hierarchy of nodes with a root.
* **Types**:
  + Binary Tree.
  + Binary Search Tree (BST).
  + AVL Tree.
* **Applications**:
  + File Systems.
  + Decision Trees.

**6. Graphs**

* **Definition**: A set of nodes (vertices) connected by edges.
* **Types**:
  + Directed and Undirected Graphs.
  + Weighted and Unweighted Graphs.
* **Applications**:
  + Social Networks.
  + Shortest Path Algorithms (e.g., Dijkstra’s Algorithm).

**7. Algorithms Using Data Structures**

* **Sorting Algorithms**: Bubble Sort, Quick Sort, Merge Sort (use arrays).
* **Search Algorithms**: Binary Search, Depth-First Search (DFS), Breadth-First Search (BFS).

**References for Further Reading**

1. *Introduction to Algorithms* by Cormen, Leiserson, Rivest, Stein.
2. *Data Structures and Algorithms Made Easy* by Narasimha Karumanchi.
3. *The Algorithm Design Manual* by Steven Skiena.