Introduction to Linked List: List: The Building Blocks of of Data Structures

- A linked list is a linear data structure where elements are stored in separate nodes.
  Learn about its advantages, types, operations, complexities, and applications.

Advantages and Disadvantages of Linked Lists Linked Lists

1 **Advantages**

Dynamic size, efficient insertion and deletion, flexibility of structure.

2 Disadvantages

Slower access time, extra memory for storing pointers.

Types of Linked Lists

Singly Linked List

Each node contains data and a reference to the next node.
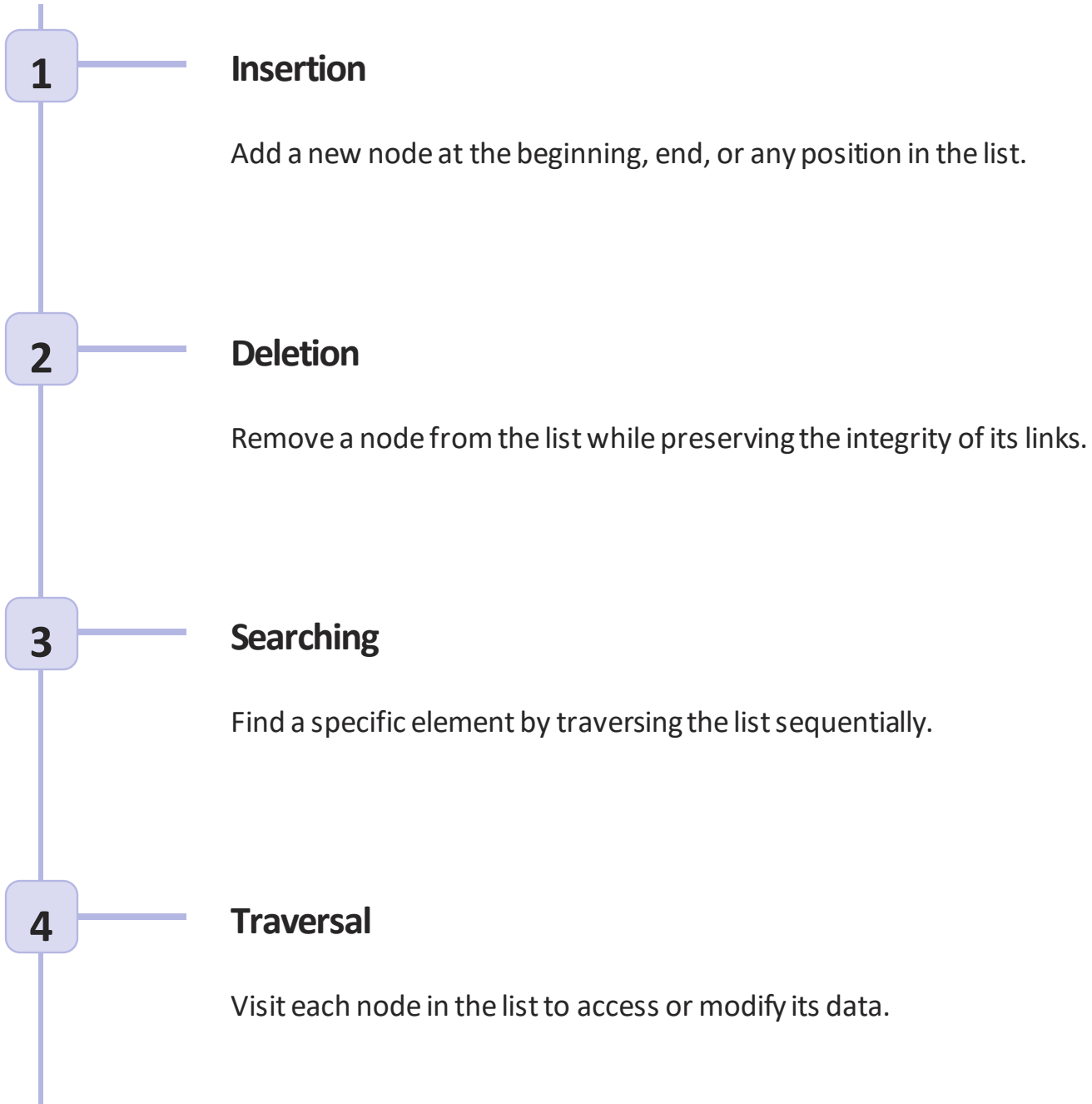
Doubly Linked List

Nodes have references to the previous and next nodes.

Circular Linked List

Last node points back to the first node, creating a circular structure.

Basic Operations on a Linked List

1 **Insertion**

Add a new node at the beginning, end, or any position in the list.

2 **Deletion**

Remove a node from the list while preserving the integrity of its links.

3 **Searching**

Find a specific element by traversing the list sequentially.

4 **Traversal**

Visit each node in the list to access or modify its data.

Complexity Analysis of Linked List Operations

Insertion

Best case: O(1), Worst case: O(n), Average case: O(n)

Deletion

Best case: O(1), Worst case: O(n), Average case: O(n)

Searching

Best case: O(1), Worst case: O(n), Average case: O(n)

Traversal

Best case: O(1), Worst case: O(n), Average case: O(n)

Applications of Linked Lists

1 **Stacks and Queues**

Used as underlying data structures for implementing stack and queue operations.

2 **Graphs**

Adjacency lists can be implemented using linked lists to represent graph structures.

3 **Dynamic Memory Allocation**

Linked lists are used to manage memory dynamically by allocating and deallocating memory chunks.