1. Understand Linked Lists:

o Explain the different types of linked lists (Singly Linked List, Doubly Linked List).

Linked lists are data structures that consist of nodes connected by references (or pointers). Each node contains data and a reference to the next node in the list.

Types of Linked Lists:

Singly Linked List:

Structure: Each node contains a data element and a reference (or link) to the next node in the sequence. The last node points to null.

Advantages:

Simplicity: Simple structure with one reference per node.

Efficient Insertions/Deletions: Inserting or deleting a node is efficient if the position is known, as only the references need to be updated.

Disadvantages:

Unidirectional Traversal: Can only traverse in one direction (forward).

Doubly Linked List:

Structure: Each node contains a data element, a reference to the next node, and a reference to the previous node. The first node's previous reference and the last node's next reference point to null.

Advantages:

Bidirectional Traversal: Can traverse in both forward and backward directions.

Flexible Deletions: Easier to delete a node when the previous node is also referenced.

Disadvantages:

Increased Memory Usage: Requires additional space for the previous node reference.

4. Analysis:

o Analyze the time complexity of each operation

Time Complexity of Each Operation:

Add Task:

Time Complexity: O(n) – Adding a task involves traversing to the end of the list if the list is not empty. Inserting at the end requires O(n) time in the worst case.

Search Task:

Time Complexity: O(n) – Searching for a task involves traversing the list from the head to the end in the worst case.

Traverse Tasks:

Time Complexity: O(n) – Traversing the entire list involves visiting each node exactly once.

Delete Task:

Time Complexity: O(n) – Deleting a task involves searching through the list, which takes O(n) time. After locating the node, deletion involves updating references, which is O(1).

o Discuss the advantages of linked lists over arrays for dynamic data.

Advantages of Linked Lists Over Arrays:

Dynamic Size:

Advantage: Linked lists can dynamically grow and shrink in size, making them more flexible for managing collections with unknown or variable sizes.

Efficient Insertions/Deletions:

Advantage: Insertion and deletion operations can be more efficient in a linked list, especially when done at the beginning or middle of the list, as only references need to be updated.

No Need for Resizing:

Advantage: Unlike arrays, linked lists do not require resizing or copying of elements when growing or shrinking.