**1.Understand Linked Lists:**

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

Singly Linked List:

Description: A singly linked list is a type of data structure that consists of nodes. Each node contains a data field and a reference (or link) to the next node in the sequence. The list terminates when a node points to null.

Advantages:

Dynamic size: Can grow or shrink as needed.

Efficient insertions/deletions: Can easily insert or delete nodes without reorganizing the entire data structure.

Doubly Linked List:

Description: A doubly linked list is similar to a singly linked list, but each node contains an additional reference to the previous node. This allows traversal in both directions (forward and backward).

Advantages:

Bidirectional traversal: Can traverse the list in both directions.

Easier deletions: Deletion operations are easier since we have access to the previous node.

**2.Analysis:**

***Q1.Analyze the time complexity of each operation.***

Time Complexity:

Add:

Best Case: O(1) (if adding at the head)

Average Case: O(n) (if adding at the end)

Worst Case: O(n) (if adding at the end)

Search:

Best Case: O(1) (if the element is at the head)

Average Case: O(n)

Worst Case: O(n) (if the element is at the end or not found)

Traverse:

Best Case: O(n)

Average Case: O(n)

Worst Case: O(n)

Delete:

Best Case: O(1) (if deleting the head)

Average Case: O(n)

Worst Case: O(n) (if deleting the last element or not found)

Advantages of Linked Lists over Arrays for Dynamic Data:

***Q2.Discuss the advantages of linked lists over arrays for dynamic data.***

Dynamic Size: Linked lists can grow or shrink in size as needed, unlike arrays which have a fixed size.

Efficient Insertions/Deletions: Inserting or deleting nodes in a linked list is more efficient as it doesn't require shifting elements, as in arrays.

Memory Allocation: Linked lists use dynamic memory allocation, which means they can utilize memory more efficiently without requiring a contiguous block of memory.