**Understand Linked Lists:**

**Q1. Explain the different types of linked lists (Singly Linked List, Doubly Linked List).**

Answer:

**Linked Lists Explanation**

A linked list is a linear data structure where each element is a separate object, known as a node. Each node contains two items: the data and a reference (or link) to the next node in the sequence. This structure allows for efficient insertion or removal of elements from any position in the sequence.

**Singly Linked List**

A singly linked list is a type of linked list where each node only has a reference to the next node in the sequence.

**Doubly Linked List**

A doubly linked list is a type of linked list where each node has references to both the next and previous nodes in the sequence.

**Analysis:**

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

Answer: **Time Complexity Analysis**

| **Operation** | **Time Complexity** |
| --- | --- |
| Add Task | O(n) |
| Search Task | O(n) |
| Traverse Tasks | O(n) |
| Delete Task | O(n) |

Note: n is the number of tasks in the linked list.

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

Answer: **Advantages of Linked Lists over Arrays**

1. **Dynamic Memory Allocation**: Linked lists can grow or shrink dynamically as elements are added or removed, whereas arrays have a fixed size.
2. **Efficient Insertion and Deletion**: Linked lists allow for efficient insertion and deletion of elements at any position, whereas arrays require shifting elements which can be time-consuming.
3. **Good Memory Usage**: Linked lists use memory efficiently, as each node only allocates memory for the data and the reference to the next node.

However, linked lists have some disadvantages, such as slower search times and more complex implementation compared to arrays. Ultimately, the choice between linked lists and arrays depends on the specific requirements of the problem.