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

**A.** Singly Linked List:

* A singly linked list is a collection of nodes where each node contains data and a reference (or link) to the next node in the sequence.
* The list starts with a head node, which is the first node in the list.
* The last node points to null, indicating the end of the list.

Doubly Linked List:

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

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

**A.** Time Complexity:

* Add Task: O(n) (as we need to traverse to the end of the list to add a new task)
* Search Task: O(n) (as we need to traverse the list to find a task)
* Traverse Tasks: O(n) (as we traverse the entire list)
* Delete Task: O(n) (as we need to find the task before deleting it)

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

**A.** Advantages of Linked Lists over Arrays for Dynamic Data:

* Dynamic Size: Linked lists can easily grow and shrink in size by adding or removing nodes, whereas arrays have a fixed size.
* Efficient Insertions/Deletions: Inserting or deleting an element in a linked list is more efficient (O(1) for inserting/deleting at the beginning) than in an array, where shifting elements is required (O(n)).
* Memory Utilization: Linked lists use memory more efficiently for dynamic data since they do not require contiguous memory allocation like arrays.