**Task Management System**

1. **Explain different types of Linked lists.**

* There are mainly two types of linked lists: Singly Linked Lists and Doubly Linked Lists. In a Singly Linked List, each node contains data and a reference (or pointer) to the next node in the sequence. This structure is simple and memory-efficient, making it ideal for basic operations such as insertion or deletion at the head or tail of the list. However, it doesn't support backward traversal.
* On the other hand, a Doubly Linked List enhances functionality by including two references in each node one pointing to the next node and another to the previous one. This bidirectional nature makes operations like reverse traversal and deletion of a middle node more efficient. While doubly linked lists consume slightly more memory due to the extra pointer, they offer greater flexibility for complex operations.

1. **Analysis of Time complexity of each operation.**

Add:

* O(n) — Traverses to the end to add the task.

Search:

* O(n) — Linear search through the list.

Traverse:

* O(n) — Visits each node to display the tasks.

Delete (by ID):

* O(n) — Searches for the node and relinks pointers.

1. **Advantages of LinkedLists over Arrays for dynamic data.**

* Dynamic Size:  
   Linked lists can grow or shrink at runtime without needing to reallocate or resize memory.
* Efficient Insertion/Deletion:  
   Elements can be added or removed easily, especially at the head or middle, without shifting others.
* Memory Utilization:  
   Memory is allocated only when needed, unlike arrays that may reserve extra unused space.
* No Waste of Space:  
   Ideal for scenarios with unknown or changing data size, as there's no fixed-size limitation.