## Task Management System

1. **Explain the different types of linked lists:**

* A **Singly Linked List** is a data structure where each node contains data and a reference (or pointer) to the next node in the sequence. It allows traversal in only one direction—from the head to the end of the list.
* A **Doubly Linked List** extends this by having each node contain references to both the next and the previous nodes, allowing traversal in both directions, forward and backward.

1. Time Complexity Analysis

* **Add:** Adding a task to the beginning of a singly linked list takes constant time, **O(1)**, because it only involves updating pointers without traversing the list and for adding anywhere else in the list it takes **O(n)**.
* **Search:** Searching for a task by its ID requires checking each node one by one until the task is found or the list ends, so it takes linear time, **O(n)**.
* **Traverse:** Traversing the linked list to visit and display all tasks requires going through every node, which takes linear time, **O(n)**.
* **Delete:** Deleting a task involves searching for the node and then updating pointers to remove it, which takes linear time, **O(n)**, because you may need to traverse the list to find the task before deleting it.

1. Advantages of Linked Lists over Arrays

* Linked lists provide advantages over arrays for dynamic data because they allow easy insertion and deletion of elements without the need to shift other elements, and they can grow or shrink in size during runtime without memory reallocation