**Task Management System**

**Understand Linked Lists**

* **Explain the different types of linked lists (Singly Linked List, Doubly Linked List).**
* **Singly Linked List:**
* **Structure:** Each node contains data and a reference (link) to the next node.
* **Traversal:** Can only be traversed in one direction (forward).
* **Usage:** Suitable for simple sequential access where backward traversal is not needed.
* **Doubly Linked List:**
* **Structure:** Each node contains data, a reference to the next node, and a reference to the previous node.
* **Traversal:** Can be traversed in both directions (forward and backward).
* **Usage:** Ideal for scenarios requiring bidirectional traversal and easier deletion/insertion of nodes.

**Analysis**

* **Analyse the time complexity of each operation.**
* **Time Complexity Analysis:**
* **Adding a Task:** In a singly linked list, appending a task at the end requires traversing the entire list to reach the last node, resulting in time complexity, where n is the number of nodes.
* **Searching for a Task:** Finding a task by its ID necessitates scanning through the list, leading to a worst-case time complexity of .
* **Traversing Tasks:** Displaying all tasks involves visiting each node once, which takes time.
* **Deleting a Task:** Removing a task involves first locating it, which is an ) operation. The actual removal of the node is , but the overall time complexity remains O(n) due to the search process.
* **Discuss the advantages of linked lists over arrays for dynamic data.**
* **Dynamic Size:**
* Linked lists can easily grow and shrink in size without the need for resizing or reallocating memory.
* **Efficient Insertions/Deletions:**
* Adding or removing elements is efficient (time) if the position is known, as it only involves updating pointers.