**Task Management System – Analysis:**

### Understanding Linked Lists :

1. **Singly Linked List:** A singly linked list is a collection of nodes where each node contains data and a reference (or pointer) to the next node in the sequence.
   * **Operations:**
     + **Insertion:** O(1) at the beginning, O(n) at the end (if traversal is required).
     + **Deletion:** O(1) if node to delete is known; otherwise, O(n) for traversal.
     + **Traversal:** O(n), as each node needs to be visited sequentially.
2. **Doubly Linked List:** A doubly linked list is similar to a singly linked list but with an additional reference to the previous node, allowing traversal in both directions.
   * **Operations:**
     + **Insertion:** O(1) if node to insert before/after is known.
     + **Deletion:** O(1) if node to delete is known.
     + **Traversal:** O(n), but can traverse in both forward and backward directions.
3. **Analysis of Time Complexity for Task Management System**

* **Add Task:** O(n) in the worst case (if traversing to find the end is required). Otherwise, O(1) if inserting at the head.
* **Search Task:** O(n) as it requires traversal from the head to the end.
* **Delete Task:** O(n) if traversal is needed to find the node. Deletion itself is O(1) once the node is located.
* **Traverse Tasks:** O(n) as each node must be visited sequentially.

1. **Advantages of Linked Lists over Arrays**

* **Dynamic Size:** Linked lists can easily grow or shrink in size by dynamically allocating and deallocating memory. Arrays require a fixed size or expensive resizing operations.
* **Efficient Insertions/Deletions:** Insertions and deletions are more efficient in linked lists, especially if performed at the beginning or middle of the list. In arrays, these operations often require shifting elements, which can be costly.
* **Memory Utilization:** Linked lists can make better use of memory when dealing with unknown or variable sizes, as they allocate memory only as needed.