**Exercise 5: Task Management System**

**Scenario:**

You are developing a task management system where tasks need to be added, deleted, and traversed efficiently.

**Step 1: Understand Linked Lists**

**Singly Linked List**

* Concept: A singly linked list is a data structure that consists of nodes. Each node contains data and a pointer to the next node in the sequence. The list starts with a head node and ends with a node pointing to null.
* Operations: Adding, deleting, and traversing elements are typical operations.

**Doubly Linked List**

* Concept: A doubly linked list is similar to a singly linked list but each node contains two pointers: one to the next node and one to the previous node.
* Operations: It allows traversal in both directions (forward and backward), making operations like deletion easier.

**Step2: Setup**

Create a Task Class:Define a Task class with attributes taskId, taskName, and status.

**Step 3: Implementation**

Singly Linked List to Manage Tasks: Define the node structure and linked list operations**.**

**Step 4: Analysis**

**Time Complexity**

* Add Task: O(n)- In the worst case, we traverse the entire list to add a new task at the end.
* Search Task: O(n) - We might need to check each task to find the target task.
* Delete Task: O(n) - We might need to traverse the entire list to find and delete the target task.
* Traverse Tasks: O(n) - We need to visit each task once.

**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. Arrays have a fixed size, requiring resizing when the capacity is exceeded, which can be costly.
* Ease of Insertion/Deletion: Inserting or deleting nodes in a linked list does not require shifting elements as in arrays. This makes linked lists more efficient for operations involving frequent additions and deletions.
* Memory Utilization: Linked lists allocate memory as needed, while arrays allocate memory in advance, which may lead to wasted space if the array is not fully utilized.