**Exercise 5: Task Management System**

**1. Understand Linked Lists**

* **Singly Linked List**: Consists of nodes where each node points to the next node. It's efficient for insertions and deletions at the head of the list, but requires linear time to search for elements. It supports one-way traversal.
* **Doubly Linked List**: Each node points to both the next and previous nodes. This allows traversal in both directions and makes insertions and deletions more flexible, but uses more memory due to the additional pointer and requires more complex operations.

**2. Setup**

* **Task Class**: A class Task represents each task with attributes like taskId, taskName, and status, which are used to manage and track individual tasks within the system.

**3. Implementation**

* **Singly Linked List**:
  + **Add**: Insert a new task at the beginning or end of the list.
  + **Search**: Locate a task by its ID by traversing the list.
  + **Traverse**: Go through each task in the list to view all tasks.
  + **Delete**: Remove a task by ID, which may require traversing the list to find the task.

**4. Analysis**

* **Time Complexity**:
  + **Add**: O(1) - Inserting at the head is constant time.
  + **Search**: O(n) - Requires linear time to find a task.
  + **Traverse**: O(n) - Each node is visited once.
  + **Delete**: O(n) - Finding and removing a task requires linear time.
* **Advantages of Linked Lists Over Arrays**:
  + **Dynamic Size**: Linked lists grow and shrink dynamically, while arrays have a fixed size.
  + **Efficient Insertions/Deletions**: Easier and more efficient to insert or delete tasks compared to shifting elements in an array.
  + **Memory Usage**: Better suited for systems where the number of tasks changes frequently, avoiding the need for resizing or wasting memory.