**Exercise 5: Task Management System**

**Scenario:**

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

**Understanding Linked Lists**

**Singly Linked List:**  
A singly linked list consists of nodes where each node has two parts: data and a reference to the next node in the sequence. Traversal is done in one direction only.

**Doubly Linked List:**  
A doubly linked list is similar to a singly linked list but with an additional reference in each node that points to the previous node. This allows for traversal in both directions (forward and backward).

**Analysis**

**Time Complexity of Operations**

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

**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 and require resizing if the capacity is exceeded.
* **Efficient Insertions/Deletions:** Insertions and deletions in linked lists can be done efficiently (O(1) time) if the position is known. In arrays, these operations require shifting elements, which can be expensive (O(n) time).
* **Memory Usage:** Linked lists allocate memory for each element dynamically, whereas arrays allocate memory for all elements upfront. This can lead to wasted space in arrays if the allocated size is not fully utilized.