**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 linked list where each node contains a reference (or link) to the next node in the sequence. It allows traversal in one direction.
* **Doubly Linked List:** A linked list where each node contains references to both the next and the previous node. This allows traversal in both direction

**Analysis:**

* **Add Task:**
  + Time Complexity: O(n) (because you need to traverse to the end of the list to add a new task).
  + Space Complexity: O(1).
* **Search Task:**
  + Time Complexity: O(n) (because you need to traverse the list to find the task).
  + Space Complexity: O(1).
* **Traverse Tasks:**
  + Time Complexity: O(n) (because you need to traverse the entire list).
  + Space Complexity: O(1).
* **Delete Task:**
  + Time Complexity: O(n) (because you need to find the task to delete).
  + Space Complexity: O(1).

**Advantages of Linked Lists over Arrays:**

* **Dynamic Size:** Linked lists can grow and shrink dynamically, whereas arrays have a fixed size.
* **Efficient Insertions/Deletions:** Inserting or deleting a node in a linked list is more efficient than in an array, especially when dealing with large data sets, as it does not require shifting elements.
* **Memory Utilization:** Linked lists utilize memory more efficiently than arrays, as they allocate memory as needed.

**Disadvantages of Linked Lists:**

* **Memory Overhead:** Each node in a linked list requires additional memory for storing the reference to the next node.
* **Sequential Access:** Linked lists do not allow random access to elements, making operations like search slower compared to arrays.