**Exercise 5: Task Management System**

**Scenario:**

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

Sure, here are the answers in a concise and humanized mode:

**Different Types of Linked Lists**

1. Singly Linked List:

Structure: Each node points to the next node.

Traversal: Only forward traversal is possible.

2. Doubly Linked List:

Structure: Each node points to both the next and the previous node.

Traversal: Allows traversal both forward and backward.

**Time Complexity of Each Operation**

1. Add Task: O(n)

Reason: We traverse to the end of the list to add a new task.

Optimization: Keeping a reference to the tail could make this O(1).

2. Search Task: O(n)

Reason: We may need to check each node to find the task.

3. Traverse Tasks: O(n)

Reason: We visit each node to print tasks.

4. Delete Task: O(n)

Reason: We need to find the task, then adjust pointers.

5. Update Task: O(n)

Reason: Similar to search, we need to find the task first.

**Advantages of Linked Lists Over Arrays for Dynamic Data**

1. Dynamic Size:

Linked Lists: Easily grow and shrink as needed.

-Arrays: Fixed size once created, resizing requires creating a new array.

2. Efficient Insertions/Deletions:

Linked Lists: Insertions and deletions are efficient as we only adjust pointers.

Arrays: Inserting/deleting requires shifting elements, which is less efficient.