1. **Singly Linked List**: Each node contains data and a reference to the next node in the sequence. It allows traversal in one direction.

**Doubly Linked List**: Each node contains data, a reference to the next node, and a reference to the previous node. It allows traversal in both directions.

4. **Time Complexity**:

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

**Advantages of Linked Lists over Arrays for Dynamic Data**:

* Dynamic Size: Linked lists can easily grow and shrink in size by allocating or deallocating nodes, whereas arrays have a fixed size.
* Efficient Insertions/Deletions: Inserting or deleting elements in a linked list is more efficient than in an array, especially if the operations occur at the beginning or middle of the list. In arrays, these operations require shifting elements, leading to O(n) time complexity.