Exercise 5: Task Management System

Types of Linked Lists

Singly Linked List:

Each node contains data and a reference to the next node.

Traversal is unidirectional, from the head to the last node (tail).

Operations:

Insert: O(1) if inserting at the head, O(n) if inserting at the end.

Delete: O(1) if deleting at the head, O(n) for other positions.

Search: O(n) on average.

Doubly Linked List:

Each node contains data, a reference to the next node, and a reference to the previous node.

Traversal is bidirectional, both forward and backward.

Operations:

Insert: O(1) if inserting at the head or tail.

Delete: O(1) if deleting at the head or tail, O(n) for other positions.

Search: O(n) on average.

Time Complexity

Add Task: O(n) (to add at the end)

Search Task: O(n) (linear search)

Traverse Tasks: O(n)

Delete Task: O(n) (to find the task)

Advantages of Linked Lists Over Arrays for Dynamic Data

Dynamic Size: Linked lists can grow and shrink dynamically without needing to resize or allocate additional memory.

Efficient Insertions/Deletions: Insertions and deletions are more efficient as they generally require only changing a few references/pointers, whereas arrays may require shifting elements.

Memory Utilization: Linked lists allocate memory as needed, which can be more memory efficient compared to arrays that may allocate more memory than necessary.