**Exercise 5: Task Management System**

**Explain the different types of linked lists (Singly Linked List, Doubly Linked List).**

Singly Linked List:

Consists of nodes where each node has two components: data and a reference (or pointer) to the next node in the sequence.

Advantages:

Dynamic size, allowing for efficient memory utilization.

Easy insertion and deletion of nodes (especially at the beginning).

Disadvantages:

No backward traversal.

More memory usage due to storing references.

Doubly Linked List:

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

Advantages:

* Allows for traversal in both directions (forward and backward).
* Easier deletion of nodes since each node has a reference to the previous node.

Disadvantages:

* Increased memory usage due to storing two references per node.
* Slightly more complex implementation.

**Analyze the time complexity of each operation.**

Add Operation:

* Best Case: O(1) - Adding at the beginning of the list.
* Worst Case: O(n) - Adding at the end of the list.

Search Operation:

* Best Case: O(1) - When the element to be searched is the first element.
* Average Case: O(n) - On average, the element is somewhere in the middle.
* Worst Case: O(n) - When the element is the last or not present.

Traverse Operation:

* Best Case: O(n) - Always O(n) since all elements need to be visited.
* Worst Case: O(n) - Always O(n) since all elements need to be visited.

Delete Operation:

* Best Case: O(1) - When the element to be deleted is the first element.
* Average Case: O(n) - Requires searching for the element and then deleting it.
* Worst Case: O(n) - When the element to be deleted is the last element or not present.

**Discuss the advantages of linked lists over arrays for dynamic data.**

* Linked lists can grow and shrink in size dynamically, allowing for efficient use of memory.
* Insertions and deletions can be done easily and efficiently without the need to shift elements, as required in arrays.
* Linked lists do not need a contiguous block of memory, which can lead to better memory utilization for large datasets.