**Exercise 5: Task Management System**

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

**Types of Linked Lists:**

1. **Singly Linked List:** Each node contains a data part and a pointer to the next node. The list is traversed in one direction, from the head to the end.
2. **Doubly Linked List:** Each node contains a data part, a pointer to the next node, and a pointer to the previous node. This allows traversal in both forward and backward directions.

**2) Analyse the time complexity of each operation, Add, Search, Traverse, Delete (Linked List)**

1. **Add:**

* **At the beginning:** O (1)
  + - Adding a new node at the beginning involves updating the head pointer, which takes constant time.
* **At the end:** O(n)
  + - Adding a new node at the end requires traversing the entire list to find the last node, which takes linear time.
* **At a specific position:** O(n)
  + - Adding at a specific position involves traversing the list to that position, which takes linear time.

1. **Search:**

* **O(n)**: Searching for a specific element requires checking each node one by one, which takes linear time.

1. **Traverse:**

* **O(n)**: Traversing the entire linked list involves visiting each node once, which takes linear time.

1. **Delete:**

* **At the beginning:** O (1)

Deleting the first node involves updating the head pointer, which takes constant time.

* **At the end:** O(n)
  + - Deleting the last node requires traversing the entire list to find the second-last node, which takes linear time.
* **At a specific position:** O(n)
  + - Deleting at a specific position involves traversing the list to that position, which takes linear time.

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

Linked lists are better than arrays for dynamic data because they can easily grow and shrink in size. With linked lists, adding or removing elements is simple and efficient, as it only involves changing pointers. This makes them ideal for situations where the size of the data set changes frequently. In contrast, arrays have a fixed size, and resizing them requires creating a new array and copying all the elements, which is time-consuming. Additionally, linked lists don't use unnecessary memory since they only use as much space as needed, unlike arrays which may have unused elements.