**Different types of linked lists:**

1. **Singly Linked List:** A singly linked list is a data structure in which each element points to the next element in the list. Each element is a separate object, and each element (called a "node") points to the next node in the sequence.
2. **Doubly Linked List:** A doubly linked list is a data structure in which each element points to both the next element and the previous element in the list. This allows for efficient insertion and deletion of elements at any position in the list.

**Advantages of linked lists:**

1. **Dynamic memory allocation:** Linked lists can allocate memory dynamically, which means that the memory is allocated at runtime. This makes them suitable for applications where the number of elements is unknown at compile-time.
2. **Efficient insertion and deletion:** Linked lists allow for efficient insertion and deletion of elements at any position in the list, which makes them suitable for applications where elements are frequently added or removed.

**The time complexity of each operation:**

1. **Add operation:** The time complexity of the add operation is O(n) because it iterates through the entire linked list to add a new task.
2. **Search operation:** The time complexity of the search operation is O(n) because it iterates through the entire linked list to find a task by ID.
3. **Traverse operation:** The time complexity of the traverse operation is O(n) because it iterates through the entire linked list to print task information.
4. **Delete operation:** The time complexity of the delete operation is O(n) because it iterates through the entire linked list to find and delete a task by ID.

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

Linked lists have several advantages over arrays for dynamic data, including:

1. **Dynamic memory allocation:** Linked lists can allocate memory dynamically, which makes them suitable for applications where the number of elements is unknown at compile-time.
2. **Efficient insertion and deletion:** Linked lists allow for efficient insertion and deletion of elements at any position in the list, which makes them suitable for applications where elements are frequently added or removed.
3. **Flexible data structure:** Linked lists are a flexible data structure that can be used to implement stacks, queues, and other