**Exercise 5: Task Management System**

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

* **Singly Linked List (SLL)**:
* **Structure**: Each node contains data and a reference (or pointer) to the next node in the sequence.
* **Characteristics**: Unidirectional traversal; you can only move forward through the list.
* **Use Cases**: Simpler implementation, useful when you only need to traverse in one direction.
* **Doubly Linked List (DLL)**:
* **Structure**: Each node contains data, a reference to the next node, and a reference to the previous node.
* **Characteristics**: Bidirectional traversal; you can move forward and backward through the list.
* **Use Cases**: More complex than SLLs, but useful when bidirectional traversal is needed.

1. **Analysis:**
   * *Analyze the time complexity of each operation.*

* **Add Task**: O(n) (traversing to the end of the list)
* **Search Task**: O(n) (traversing the list to find the task)
* **Delete Task**: O(n) (traversing the list to find the task and update links)
* **Traverse Tasks**: O(n) (traversing the entire list)
  + *Discuss the advantages of linked lists over arrays for dynamic data.*
* **Dynamic Size**: Linked lists can grow or shrink as needed, unlike arrays with fixed size.
* **Efficient Insertions/Deletions**: Adding or removing elements in linked lists does not require shifting elements, making these operations more efficient, especially in the middle of the list.
* **Memory Usage**: Linked lists do not need contiguous memory blocks like arrays, reducing memory fragmentation issues.