**Exercise 5: Task Management System**

**1. Understand Linked Lists**

**Singly Linked List**

* **Description**: A singly linked list is a linear data structure where each element, called a node, contains data and a reference (or link) to the next node in the sequence.
* **Advantages**:
  + **Dynamic Size**: Can grow and shrink in size by allocating and deallocating memory during runtime.
  + **Efficient Insertion/Deletion**: Insertion and deletion of nodes can be done efficiently, especially when dealing with the head or tail of the list.

**Doubly Linked List**

* **Description**: A doubly linked list is similar to a singly linked list but each node contains two references: one to the next node and another to the previous node.
* **Advantages**:
  + **Bidirectional Traversal**: Can be traversed in both forward and backward directions.
  + **Efficient Deletion**: Easier to delete nodes when the reference to the previous node is available.

**4. Analysis**

**Time Complexity of Operations**

* **Add**: O(n) - Adding a task at the end of the list requires traversing the entire list in the worst case.
* **Search**: O(n) - Searching for a task requires traversing the list.
* **Traverse**: O(n) - Traversing the list requires visiting each node once.
* **Delete**: O(n) - Deleting a task requires searching for it first, which involves traversing the list.

**Advantages of Linked Lists Over Arrays for Dynamic Data**

* **Dynamic Size**: Linked lists can grow and shrink as needed, whereas arrays have a fixed size.
* **Efficient Insertions/Deletions**: Inserting or deleting elements in a linked list is more efficient, especially at the beginning or middle of the list, as it doesn't require shifting elements as in an array.
* **Memory Utilization**: Linked lists utilize memory more efficiently for dynamic data as they only use as much memory as needed for the current number of elements.