**EXERCISE 5 : Task Management System**

**Understanding Linked Lists**

**Linked Lists Overview:**

A linked list is a data structure consisting of a sequence of elements, where each element (node) contains a reference (or link) to the next element in the sequence.

**Types of Linked Lists:**

1. **Singly Linked List:**
   * **Structure:** Each node contains data and a reference to the next node. It allows traversal in one direction.
   * **Advantages:** Simplicity and efficient insertion/deletion at the head.
   * **Disadvantages:** Cannot traverse backward; searching for a node requires O(n) time.
2. **Doubly Linked List:**
   * **Structure:** Each node contains data, a reference to the next node, and a reference to the previous node. It allows traversal in both directions.
   * **Advantages:** Efficient traversal both forwards and backwards; efficient insertion/deletion at both ends.
   * **Disadvantages:** More memory usage due to the extra reference (previous pointer).

**Analysis**

**Time Complexity:**

* **Add Task:**
  + **Time Complexity:** O(n) in the worst case when the list is traversed to add at the end.
  + **Explanation:** If adding at the end, we may need to traverse the entire list.
* **Search Task:**
  + **Time Complexity:** O(n)
  + **Explanation:** Searching requires traversing the list, which can be up to O(n) in the worst case.
* **Traverse Tasks:**
  + **Time Complexity:** O(n)
  + **Explanation:** Traversing involves visiting each node once.
* **Delete Task:**
  + **Time Complexity:** O(n)
  + **Explanation:** Deleting involves searching for the node to delete, which takes O(n) time.

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

1. **Dynamic Size:** Linked lists can easily grow and shrink in size without the need for resizing or reallocating memory, unlike arrays which have a fixed size.
2. **Efficient Insertions and Deletions:** Inserting or deleting nodes, especially at the beginning or middle, is efficient and does not require shifting elements, as is necessary with arrays.
3. **No Wasted Space:** Linked lists do not require allocating extra space as arrays do for resizing or unused elements.