Data Structures and Algorithms

Exercise 5:

Task Management System

This project implements a Task Management System using Java and a singly linked list. It supports efficient task addition, deletion, searching, and traversal while demonstrating dynamic memory handling. Here's a detailed explanation:

Step 1: Understand Linked Lists

- **Singly Linked List:** Each node points to the next.
- **Doubly Linked List:** Nodes point both forward and backward (not used here).

Step 2: Setup

- *Task* class with *taskId*, *taskName*, and *status*.

Step 3: Implementation

- Node structure created for Linked List.
- Methods implemented:
    - *addTask()* – at end
    - *searchTask()* – by ID
    - *displayTasks()* – traversal
    - *deleteTask()* – by ID

Step 4: Time Complexity Analysis

Operation	Time Complexity
Add	O(n)
Search	O(n)
Traverse	O(n)
Delete	O(n)

- **Linked lists** are ideal for dynamic datasets where frequent insertions/deletions occur without memory reallocation overhead.

Output