Data Structures and Algorithms Lab

CS-F23

LAB-03

Issue Date: February 24, 2025

Start Time: 1:45 PM

Total Marks: 40 **Submission Time: 3:20 PM**

The objective of this lab is to:

This lab aims to enhance students' understanding of Abstract Data Types (ADTs) and how to efficiently manipulate Linked lists

Instructions:

1) Use only **the ADT functions** to access and modify elements.
2) Implement each function **efficiently** to minimize time and space complexity.

Task 01(Rectangle Representation) [10 Marks]

You are given a singly linked list where each node represents a vertex of a rectangle in 2D space. Each node contains the x and y coordinates of a vertex.

Write a function to calculate the area of the rectangle using the vertices provided in the linked list.

➡ Function Prototype: ***int calculateArea(Node * head);***

Example 1

Input: (0,0) → (4,0) → (4,3) → (0,3) → null

Process:

- The given vertices represent a rectangle with opposite corners at (0,0) and (4,3).
- **Width** = Difference in x-coordinates = 4 - 0 = **4**
- **Height** = Difference in y-coordinates = 3 - 0 = **3**
- **Area** = Width × Height = **4 × 3 = 12**

Output: 12

Example 2

Input: (2,1) → (6,1) → (6,5) → (2,5) → NULL

Process:

- The given vertices represent a rectangle with opposite corners at (2,1) and (6,5).
- **Width** = Difference in x-coordinates = 6 - 2 = **4**
- **Height** = Difference in y-coordinates = 5 - 1 = **4**
- **Area** = Width × Height = **4 × 4 = 16**

Output: 16

Task 02 (Linked List Compression) [10 Marks]

You are given a singly linked list where each node contains an integer. Your task is to compress the list by replacing consecutive duplicate values with:

- A single occurrence of that value
- Followed by the count of its duplicates

If the compressed version is not shorter than the original list, return the original list instead.

➡ Function Prototype: ***Node * compressList(Node * head);***

Example :

Input: 1→ 1→ 2 → 3 → 3 → 3→ null

Output: 1 → 2 → 2 → 3 → 3 → null (where 1 → 2 means "1 appears twice").

Task 03 (Secure Message Transmission in a Spy Network) [20 Marks]

In a spy network, messages are transmitted between agents using a singly linked list.
Each node contains an encrypted value.

Due to network interference, some consecutive messages get corrupted. A corrupted message sequence is identified when consecutive values sum to zero.

The task is to remove these corrupted sequences and return the cleaned-up list.

➡ Function Prototype: ***Node * removeZeroSumSublist(Node * head);***

Example 1:

Input: 4 → 6 → −10 → 8 → 9 → −8 → -9 → 2 → null

Process:

- (4, 6, -10) sum to 0 → Remove (4, 6, -10)
- (8, 9, -8, -9) sum to 0 → Remove (8, 9, -8, -9)

Output: 2 → null

Example 2:

Input: $1 \rightarrow 2 \rightarrow 3 \rightarrow -3 \rightarrow -2 \rightarrow$ null

Process:

- (2, 3, -3, -2) sum to 0 → Remove (2, 3, -3, -2)

Output: $1 \rightarrow$ null

Good Luck!

Note: You must complete all your tasks individually. Absolutely NO collaboration is allowed. Any case of plagiarism/cheating would result in 0 marks in sessional activities.