Data Structures and Algorithms Lab

CS-F23

LAB-03

Issue Date: February 17, 2025

Start Time: 1:45 PM

Submission Time: 3:00 PM

Total Marks: 50

The objective of this lab is to:

This lab aims to enhance students' understanding of Abstract Data Types (ADTs) and how to efficiently manipulate 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(Loneliest Element) [15 Marks]

Find the "Loneliest" Element. The "loneliness" of an element in a list is defined as the sum of the absolute differences between its value and the values of all other elements in the list.

Write a function to find the element with the highest loneliness score.

➡ Function Prototype: ***int loneliest(List & l);***

Example :

Input: L = [2, 6, 8, 7, 1]

Output: Loneliest Element = 1, Loneliness Score = 19

(Explanation: The loneliness score for 1 is calculated as |1-2| + |1-6| + |1-8| + |1-7| = 1 + 5 + 7 + 6 = 19.)

Task 02 (Fold Operation) [15 Marks]

Write a function that "folds" the list in half, pairing the first element with the last, the second with the second-to-last, and so on.

If the list has an odd number of elements, leave the middle element unchanged.

➡ Function Prototype: ***void fold(List & l);***

Example :

Input: L = [2, 6, 8, 7, 1]

Output: L = [3, 13, 8]

(Explanation: Pair (2, 1) → 2 + 1 = 3, pair (6, 7) → 6 + 7 = 13, and leave 8 unchanged.)

Task 03 (Raindrop Cascade) [20 Marks]

You are provided with a **list** of integers and **k** (starting index for one initial raindrop).

Simulate the **Raindrop Cascade** operation on this list based on the following rules:

Raindrop Movement:

- Start the raindrop at the given index.
- It moves **rightward** until it reaches the **end** of the list or encounters an element **smaller** than itself.

Splash Effect:

- When a raindrop stops, it replaces the element it lands on with its own value.

Cascade Trigger:

- After landing, it **triggers** new raindrops at the **adjacent positions** (left and right), if they exist.
- These new raindrops repeat the same movement and splashing behavior

✧ *Continue this process for all triggered raindrops.*

➡ Function Prototype: ***void raindropCascade(List & l, int k);***

Example:

Input: L = [3, 2, 5, 1, 4] k = 1

Raindrop at index 1 *(value 2)*

- **Index 2** → value = **5** (greater than 2, continue).
- **Index 3** → value = **1** (smaller than 2, stop here).
- **Splash:** Replace **index 3** value **1** with **2**.
- **Updated list:** [3, 2, 5, 2, 4].
- **Triggers:** New raindrops at **index 2** and **index 4**.

✧ *Continue for index 2 and 4*

Output: [3, 2, 5, 5, 5]

--

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.