Weather Data Storage System - Assignment Report

Name: Azhar Ali
Roll No.: 2401730134
Section: B

Course: Data Structures (Course Code: ENCS205)
Programme: B.Tech - Semester 3
Session: 2025-26

Assignment Number: 01
Theme: Foundation of Data Structures

Submitted to: Department of Computer Science & Engineering

==========

1. Introduction

This report presents the implementation of a Weather Data Storage System as part of Assignment 1 for the Data Structures course. The system is designed to store, organize, and analyze temperature data using Abstract Data Types (ADTs) and 2D arrays. It also demonstrates efficient data access methods, sparse data handling, and complexity analysis.

2. Weather Record ADT

The Weather Record ADT represents a single weather entry with the following attributes: - Date: String (DD/MM/YYYY format) - City: String - Temperature: Double The ADT provides getter and setter methods to access and update these attributes, and a display function for formatted output. This design ensures data abstraction and encapsulation.

3. 2D Array Storage System

The main data storage uses a 2D array (matrix) where: - Rows represent years - Columns represent cities Each cell stores the temperature value for a given city in a given year. Missing values are handled using a sentinel value (-999.9). Additional maps (year-to-index, city-to-index) are maintained for fast O(1) access.

4. Row-Major vs Column-Major Access

Two access methods were implemented: - Row-Major: Traverses year by year, efficient for accessing all cities in one year. - Column-Major: Traverses city by city, efficient for retrieving historical data of one city. Row-Major is generally better for temporal weather analysis, while Column-Major is better for city-specific studies.

5. Sparse Data Handling

The system uses sentinel values (-999.9) to represent missing data. Sparsity is calculated as the ratio of missing records to total possible records. Recommendations: - If sparsity > 70% $\rightarrow$ Use Sparse Matrix (triplet representation). - If sparsity 30%-70% $\rightarrow$ Sentinel approach is sufficient. - If sparsity < 30% $\rightarrow$ Dense matrix is efficient.

6. Complexity Analysis

Time Complexity: - Insert: O(1) - Delete: O(1) - Retrieve: O(1) - Row/Column Access: O(rows × columns) Space Complexity: - 2D Array: O(rows × columns) - Index Maps: O(rows + columns) - Overall: Dominated by the 2D matrix.

7. Implementation Demonstration

The system was implemented in C++ with the following features demonstrated: - Creation of Weather Records and insertion into storage. - Retrieval of data for specific city and year. - Deletion of records based on city and date. - Row-major and column-major access comparisons. - Sparse data handling analysis and recommendations. - Complexity analysis and complete matrix display.

8. Conclusion

This assignment provided hands-on experience with ADTs, 2D arrays, and real-world applications of data structures. It highlighted the trade-offs between row-major and column-major access, the importance of handling sparse data, and analyzing complexity for efficient design. The Weather Data Storage System fulfills all assignment requirements successfully.