K.R. MANGALAM UNIVERSITY
THE COMPLETE WORLD OF EDUCATION

Department: CSE

Session: 2025-26

Programme: BTech, BCA

Semester: 3rd

Course Code: ENCS253

Course: Data Structures

Lab Assignment Details

Theme: Arrays and Complexity Analysis

Lab Assignment Number: 01
Total Marks: 10
Contribution: 20% of internal evaluation
CO Mapping: CO 1, CO 2

Submission Requirements:

- Submit individually via GitHub and provide the link of submission by **5th September 2025**.
- Late submissions, copied assignments, or AI-generated text will not be accepted.
- Use the dedicated lab assignment format shared in classes.

Problem: Develop the Inventory System for a Grocery Store

Description: The Inventory System for a Grocery Store is a software solution designed to store stock, track inventory status, and support operations such as restocking, low-stock alerts, and summary reports. This system leverages arrays and complexity analysis to manage item names, quantities, and prices efficiently, perform operations like insertion, deletion, and search, and optimize storage for rarely restocked products. The lab assignment focuses on implementing these functionalities in C++ or Python and analyzing their performance.

Implementation Sub-Problems

1. Implement single and multi-dimensional arrays to store item names, quantities, and price data.
2. Create functions to insert, delete, and search for items based on name or ID.
3. Use row-major and column-major ordering to manage price and quantity tables.
4. Implement sparse representation for rarely restocked products.
5. Calculate time and space complexity for each function implemented.

Evaluation Metrics

Metric	Marks	Excellent	Very Good	Satisfactory	Poor
Problem Understanding	2	2	1.5	1	0.5
Problem Solving Approach	2	2	1.5	1	0.5
Completion of the Problem	2	2	1.5	1	0.5
Participation during Class/Lab	2	2	1.5	1	0.5
Explanation of the Assignment	1	1	0.75	0.5	0.25
Attendance	1	1	0.75	0.5	0.25

Assignment Objectives

a. Develop a foundational understanding of C++/ Python programming and array-based data structures.
b. Gain practical experience with single and multi-dimensional arrays and sparse representations.
c. Implement real-world applications for inventory management in a grocery store.
d. Understand the application of row-major and column-major ordering in data organization.
e. Analyze the time and space complexity of implemented functions.

Assignment Instructions

1. Inventory Item ADT Design

Attributes:
a. ItemID: Integer, unique identifier for the item.
b. ItemName: String, name of the item.
c. Quantity: Integer, current stock quantity.
d. Price: Float, price per unit of the item.

Methods:
a. insertItem(data): Insert a new item record into the array.
b. deleteItem(ItemID): Remove an item record based on the ItemID.
c. searchItem(ItemID or ItemName): Retrieve item details based on ItemID or ItemName.

2. Inventory Management System

Attributes:
a. ItemArray: A single or multi-dimensional array to store item names, quantities, and prices.
b. SparseMatrix: A sparse representation for rarely restocked products.
c. PriceQuantityTable: A multi-dimensional array organized in row-major or column-major order.

Methods:
a. addItemRecord(): Add a new item record to the array.
b. removeItemRecord(): Delete an item record from the array.
c. searchByItem(): Search for items by name or ID.
d. managePriceQuantity(): Organize price and quantity data using row-major or column-major ordering.
e. optimizeSparseStorage(): Implement sparse representation for rarely restocked products.

Implementation Steps

1. Define the Inventory Item ADT with the specified attributes and methods using array structures in C++ or Python.
2. Implement functions to insert, delete, and search for items based on name or ID.
3. Develop row-major and column-major ordering for managing price and quantity tables.
4. Create a sparse representation for rarely restocked products to optimize storage.
5. Calculate and document the time and space complexity for each implemented function.

Evaluation Criteria

Criteria	Marks	Description
Problem Understanding	2	Demonstrates clear understanding of the problem requirements and objectives.
Problem Solving Approach	2	Effective approach to designing and implementing arrays, sparse representations, and operations in C++ or Python.
Completion of the Problem	2	Complete implementation of all required components (arrays, operations, sparse representation, complexity analysis).
Participation during Class/Lab	1	Active engagement in class/lab discussions and activities related to the lab assignment.
Explanation of the Assignment	2	Clear and detailed explanation of the implemented solution in the submission report.
Attendance	1	Consistent attendance in classes/labs relevant to the lab assignment.

Prerequisites

- Basic programming in C++ and Python.
- Understanding of arrays and control structures.

Concepts Covered

- Arrays, complexity analysis, asymptotic notations, sparse matrices.

Learning Outcomes

1. Apply array-based data structures in practical inventory management scenarios using C++ or Python.
2. Understand the use of row-major, column-major, and sparse representations in data organization.
3. Gain proficiency in implementing operations and analyzing their time and space complexity.

Submission Guidelines

1. Upload the lab assignment on GitHub and provide the link of submission by the specified deadline.
2. Ensure code is original and follows the provided format.
3. Include a detailed report covering:
   a. Description of the Inventory Item ADT.
   b. Strategy for implementing arrays, sparse representations, and operations in C++ or Python.
   c. Approach to complexity analysis for each function.
   d. Analysis of the system's efficiency and functionality.

Deliverables

- C++ or Python source code for all implemented functionalities.
- Complexity analysis report for each function.
- Test cases and output analysis.

Helpful References

1. **Books:**
   a. "Data Structures and Algorithm Analysis in C++" by Mark Allen Weiss
   b. "Data Structures and Algorithms in Python" by Michael T. Goodrich, Roberto Tamassia, and Michael H. Goldwasser
2. **Tutorials:**
   a. GeeksforGeeks: Array Data Structure
   b. GeeksforGeeks: Asymptotic Analysis
   c. Programiz: Arrays in C++ and Python