INFR 2820U: Algorithms and Data Structures

Assignment-1: Data Structures and Efficient Sorting in Online Shopping Management

Background:

Online shopping platforms require sophisticated data structures and efficient algorithms to manage large volumes of product data. This assignment explores the application of advanced data structures and sorting algorithms to enhance the performance of an online shopping platform's backend systems.

Objective:

To employ fundamental data structures for effective data management and to implement basic sorting algorithms for sorting product data.

Topics Covered:

- Fundamental Data Structures: Arrays, Linked Lists, Stacks, and Queues
- Basic Sorting Algorithms: Bubble Sort and Insertion Sort
- Algorithm Complexity Analysis

Learning Outcomes:

- Utilize fundamental data structures to store and manage data.
- Implement and understand the Bubble Sort and Insertion Sort algorithms.
- Evaluate the time complexities of these sorting algorithms in various scenarios.

Tasks:

1. Data Management Using Fundamental Structures:
   - Load and store given product data (product_data.txt) into arrays or linked lists, ensuring efficient access and management. Products attributes are: ID, Name, Price, and Category.
2. Data Manipulation Operations:
   - Insert: Efficiently add new products.
   - Update: Modify existing product details.
   - Delete: Remove products while preserving data structure integrity.
   - Search: Efficiently find products using key attributes (e.g., ID, Name).
3. Sorting Algorithm Implementation:
   - Apply Bubble or Insertion or Quick Sort to order product data by price. (No Library Function is allowed)
4. Complexity Analysis:
   - Record and compare the time taken to sort data that is already sorted, as well as data in reverse order.
   - Assess the best, average, and worst-case time complexities for sorting operations.

Deliverables: (5 marks)

A report that includes:

1. **Introduction** (Provide context and objectives for the assignment.)
2. **Snapshots of the Running Program** (2 Marks):
   - Illustrate each operation (Data Load, Insert, Update, Delete, Search, Sort)
   - Include screenshots of the script's output displaying the execution of each operations and the time required for sorting.
3. **Complexity Analysis Report** (1 Mark):
   - Elaborate on the best, average, and worst-case time complexity scenarios for sorting data within an array or LinkedList.
4. **Code Implementation** (2 Marks):
   - Provide the complete script in Python or C++ that includes the data structures and sorting algorithms.
   - Submit a public GitHub repository link of your submitted code.
5. **Conclusion** (Summarize findings, learning outcomes, and any conclusions drawn from the assignment.)

Notes: Assignments have to be completed individually. So, please avoid any sort of plagiarism. Please submit your assignment deliverables to Canvas**.**

Deadline: February 17, 2024

Contact TA:

- Lathushan Pavalavelauthan (lathushan.pavalavelauthan@ontariotechu.net)
- Utku Soyteki (utku.soytekin@ontariotechu.net)