Project Part 2: Advanced Data Structures and I/O

Due: April 23, 2025 (Late Due Date: April 29, 2025)

1 Project Description

This phase of the project extends Part 1 with two principal goals:

1. Discussing two theoretical applications related to your dataset.
2. Adapting an advanced data structure implementation to your data and performing a simple input/output task.

Note: You may work alone or in a group of two. Groups of more than two members are not permitted.

2 Project Tasks

You must complete the following two problems:

Problem 1 (50 points)

Write one paragraph each about two distinct theoretical applications related to your dataset. Each application should use a different ADT from among Stack/Queue, BST, Priority Queue, or Hash Table (Map). Note: You may use a stack *or* a queue, but not both.

Example applications using the credit-card fraud dataset: Make sure you are careful in your descriptions of ADTs and Data Structures!

- **Application 1:** Credit card records could be maintained in a self-organizing BST structure sorted on charge amount, with the main goal to have fast and efficient access to outliers in the data (i.e those that are extremely below or extremely above the average). Since the root of a self-organizing BST will have a charge amount close to the median, it should be easy to compare items with the smallest values (via In-order traversal) or largest values (via Reverse In-order traversal) to this quantity. Outlying data could then be investigated further.

- **Application 2:** Modified credit-card records could include a “risk” variable based on how likely the transaction is deemed to be fraudulent based on information like price, location, time, etc. The appropriate ADT for this data and application would then be the priority queue, with higher-risk transactions being placed higher in the queue. Investigators (manual or automated) would then “dequeue” records in order of risk when investigating potential fraud.

Problem 2

Choose one data structure implementation (excluding the list implementations) from class (e.g., ArrayStack, CLQueue, BST, Heap). Implement a C++ program with the following requirements:

1. (30 points) Read your dataset from Part 1 (or you can also use a new dataset if preferred) into your chosen structure.
2. (20 points) Print a subset of rows/records from your dataset based on user input. Some examples include (you can use any of these, modify them, or come up with your own):
    - Print out elements of a BST with a chosen attribute/column that exceeds a certain value (ex: for the CCFraud data only rows with more than an input value for amount.)
    - Print out elements from a PQ heap matching the k highest or k lowest (where k is a value input by the user) values for a chosen attribute/column.
    - Print out every kth element loaded into a CLQueue (again, where k is a value input by the user)

3 Important Notes

- Refer to or modify the main.cpp from Part 1 (or create a new one) for dataset reading and output tasks.
- Modify your chosen data structure by removing unnecessary operations (ex: there is likely no need to keep DeleteItem for the BST, and you can potentially even discard the traversals and traversal queue if you like).
- **Do not generate a new data structure from scratch or from an outside online resource – there will be a substantial penalty if it is clear your structure was not based on one that we used during class.**
- The project should **not** be submitted to Gradescope. Instead, submit it directly on Blackboard.

4 Project Outline (Required Submission)

Submit a brief report (.docx or .pdf format) containing:

- Your name (and your partner's name, if applicable).
- Chosen dataset and motivation for selection (you can replicate from Part 1 report if you choose to use the same dataset).
- Answers to Problem 1.
- Data structure chosen and modifications made in Problem 2.
- Brief description of implementation and input/output functionality in Problem 2.

5 Submission Instructions

Submit a **zip** file named `LN_FN_ProjP2.zip` containing:

- The final, preprocessed dataset in .csv format.
- Source code files (.h/.cpp), including modified data structure and main.cpp.
- Project outline report.

If working in a group of two, your submission can (and probably should) be identical on Blackboard, but clearly specify your partner in your report. Late submissions will not be accepted beyond the extended deadline (April 29, 2024).