DEPARTMENT OF COMPUTER SCIENCE
CPSC 473 / 673 - Introduction to Data Mining

Assignment 2

Due Date: Mar 14, 2025, at 11:59 PM

Notes:

- This is a **team assignment**. You will work together with your teammates to complete it.
- This assignment should be completed in either Java or Python.
- Zip all your files (code files, README, test data, **team member information**, and any other necessary files) into a single archive named “CPSC473_a#_team_name.zip” (or .rar, .7z, .tar, etc.). Submit this archive through Moodle.

Assignment Description

Once again, the main task of this assignment is straightforward: implement the FP-growth algorithm for frequent pattern mining. Include your implementation in a directory named `FPgrowth` in your submission package to Moodle. The detailed implementation requirements are mostly the same as in Assignment 1:

1. Transaction Database Input:
   a. Your program should be able to read a transaction database from a `.txt` file.
   b. The format of the transaction database file has been discussed in class.
2. Command-Line Arguments:
   a. All input parameters must be provided via command-line arguments, and the program should be executed through the command line.
   b. For example, assuming:
      i. The assignment is implemented in Java.
      ii. The executable is named `fpgrowth`.
      iii. The file path of the database is `./testdata.txt`.
      iv. The minimum support threshold is set to `50%`.
   c. The command to run your program should be:
      ```
      java fpgrowth ./testdata.txt 50
      ```
   d. Similar rules apply if you implement the program in Python.
3. README File:
   a. Provide clear instructions for compiling and running your program.
4. Program Output:
   a. Console Outputs: Your program must display the following details in the console.
      i. Minimum Support (minsup): Display the minsup value as a transaction count. For example, for `t25i10d10k.txt` with a minsup of `5%`:
      ```
      minsup = 5% = 499
      ```
      ii. Frequent Patterns Count: Display the total number of frequent patterns discovered. For example:
      ```
      |FPs| = 9
      ```
      iii. Execution Time: Display the total runtime of your program. For example:
      ```
      Total Runtime: 1.445 sec
      ```

b. File Output:
   i. Save all discovered frequent patterns to a `.txt` file named `MiningResult_<datasetFileName>.txt` (e.g., `MiningResult_t25i10d10k.txt`).
   ii. A sample output file (`MiningResult_data.txt`) is provided to demonstrate the expected format for the small dataset (`data.txt`) used in class.

Bonus Marks (exactly the same as assignment 1):

- You can earn up to 10% bonus marks for this assignment.
- To qualify for the bonus marks, document any special enhancements you implemented in your program. For example, describe techniques or optimizations, such as improvements in candidate generation, that enhance performance compared to the traditional approach.
- The document should:
  - Clearly explain what enhancements you implemented.
  - Justify why these enhancements improve performance.
- The document can be informal and does not need to be lengthy. An informal report is acceptable as long as it effectively communicates the enhancements.
- The more unique and impactful your enhancements, the better your chances of receiving full bonus marks.
- Submit the document in PDF format with the following name: `BonusMarkDocumentation_<team_name>.pdf`