Data Science HW1

Due date: **4/2 (Tue) 23:59**

TA: 張辰浩 , 資電館744

TA email: lobsterlab.cs.nthu@gmail.com

目標

- 給定 transactions 和 min support (頻率)，實作演算法找出 frequent patterns
- 可使用 python3 或是 C++
- 演算法不限，Apriori、FP Growth 等皆可
- **不得使用 frequent patterns 相關的 library**

輸入

- 輸入存有 transactions 的 txt 檔
- Item 以數字表示，範圍為 0~999
- Transactions 最多 100,000 筆
- 每筆 transaction 最多 200 個 item
- 每一行代表一筆 transaction，每筆 transaction 的 Item 之間用 “,” 區隔無空格
- 換行採用 **\n** (LF)，而不是 **\r\n** (CRLF)

輸入 (cont’d)

- input範例: sample.txt

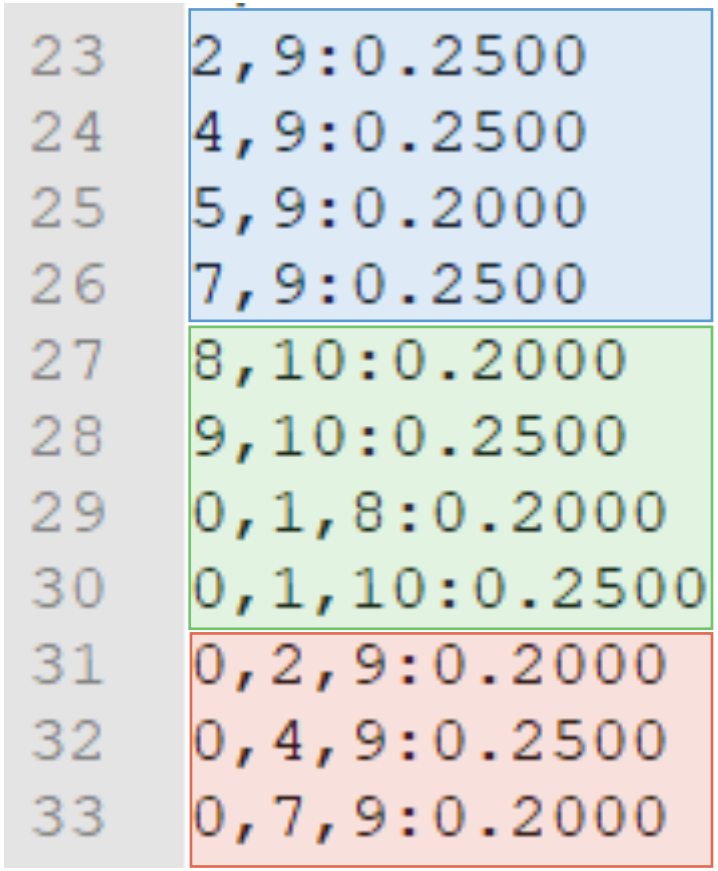
```
5,9,10
0,1,4,6,8,10
0,1,10
5
0,1,3,8,10
9
0,2,4,5,6,9
3
0,4,6,7,9,10
0,6,8,10
0,1,5,7,8,9
0,2,4,9
1,2,3,5,7,9,10
0,4,7,9
0,2,7
0,2,3,6,7,8,9
5,7,10
8
0,1,4,8,9,10
```

輸出

- 輸出一個 txt 檔
- 一行為一組 frequent pattern，frequent pattern後接上 ”:”，再接上 support (出現頻率)
  - Example: 1,2,3:0.2500
- Support 四捨五入到小數點後第4位
- 輸出的部分不需要特別排序，助教評分時會自行處理

輸出 (cont’d)

- 範例: sample.txt (min support = 0.2) 計算出的 output sample_out.txt



要求

- C++ 或 python3 擇一，程式檔名為**你的學號_hw1.cpp** or **你的學號_hw1.py**
- **不得使用 frequent patterns 相關的 library**；Python **禁止**使用 apyori、pyfpgrowth 等相關的套件，若是不確定某個 library/package能否使用，**請在eeclass提問跟助教確認**
- 在執行程式時**需在後面依序輸入3個參數**: min support、輸入檔名、輸出檔名
- **輸入輸出檔名請不要寫死！(無法順利執行以零分計)**

要求 (cont'd)

- **C++ 執行方式**
  - Compile: g++ -std=c++2a -pthread -fopenmp -O2 -o 你的學號_hw1 **你的學號_hw1.cpp**
  - Run: ./你的學號_hw1 [min support] [輸入檔名] [輸出檔名]
    (windows環境下為你的學號_hw1.exe [min support] [輸入檔名] [輸出檔名])
  - Ex: ./**12345_hw1** 0.2 input1.txt ouput1.txt
- **Python 執行方式**
  - Run: python3 **你的學號_hw1.py** [min support] [輸入檔名] [輸出檔名]

評分標準

- **共 5 筆測資**，分數根據過的筆數 (一筆全對才有分) 0~5 依序為 0、60、70、80、85、100(or 90 or 95)。
- 最後一筆測資如果輸出正確的話，會**根據速度給分**，前 33% 快的得15分，中 33% 的得 10 分，後 33% 的得 5 分。

	助教 FP Growth (C++)	Time limit
測資1	<< 1 sec	15 sec
測資2	1 sec	60 sec
測資3	2 sec	150 sec
測資4	4 sec	400 sec
測資5	30 sec	400 sec

Prejudge

- 以 **3/26 23:59** 前交的版本為主
- 可自行決定要不要參加
- 主要目的是讓大家知道**程式是否有問題**以及**執行速度**

繳交和執行環境

- C++ 或 python3 擇一繳交到 eeclass
  - **你的學號_hw1.cpp** or **你的學號_hw1.py**
- 請注意逾期**遲交不接受補交**！！
- 執行環境
  - CPU: i7-8700k
  - RAM: 32G
  - OS: Ubuntu 20.04.3 LTS
  - GPU: RTX 2080
  - gcc vesion: 11.3.0
  - Python version: python 3.9.13

For Your Reference

- 設定 VSCode
  - https://code.visualstudio.com/docs/cpp/config-mingw
  - https://code.visualstudio.com/docs/cpp/config-wsl
- https://gcc.gnu.org/projects/cxx-status.html
- https://sourceforge.net/projects/mingw-w64/
- Add **#include <climits>** if you use **INT_MAX, INT_MIN,** etc.
- Add **#include <cstring>** if you use **strcpy**, **strtok**, etc.c