

# Data Science HW1

Due date: **3/31 (Fri) 23:59**

**TA: 張辰浩 , 資電館 744**

TA email: [lobsterlab.cs.nthu@gmail.com](mailto: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，一筆 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

```
1 5, 9, 10
2 0, 1, 4, 6, 8, 10
3 0, 1, 10
4 5
5 0, 1, 3, 8, 10
6 9
7 0, 2, 4, 5, 6, 9
8 3
9 0, 4, 6, 7, 9, 10
10 0, 6, 8, 10
11 0, 1, 5, 7, 8, 9
12 0, 2, 4, 9
13 1, 2, 3, 5, 7, 9, 10
14 0, 4, 7, 9
15 0, 2, 7
16 0, 2, 3, 6, 7, 8, 9
17 5, 7, 10
18 8
19 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

23	2, 9:0.2500
24	4, 9:0.2500
25	5, 9:0.2000
26	7, 9:0.2500
27	8, 10:0.2000
28	9, 10:0.2500
29	0, 1, 8:0.2000
30	0, 1, 10:0.2500
31	0, 2, 9:0.2000
32	0, 4, 9:0.2500
33	0, 7, 9:0.2000

# 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	2.5 sec	150 sec
測資 4	6 sec	400 sec
測資 5	36 sec	400 sec

# Prejudge

- 以 **3/24 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**

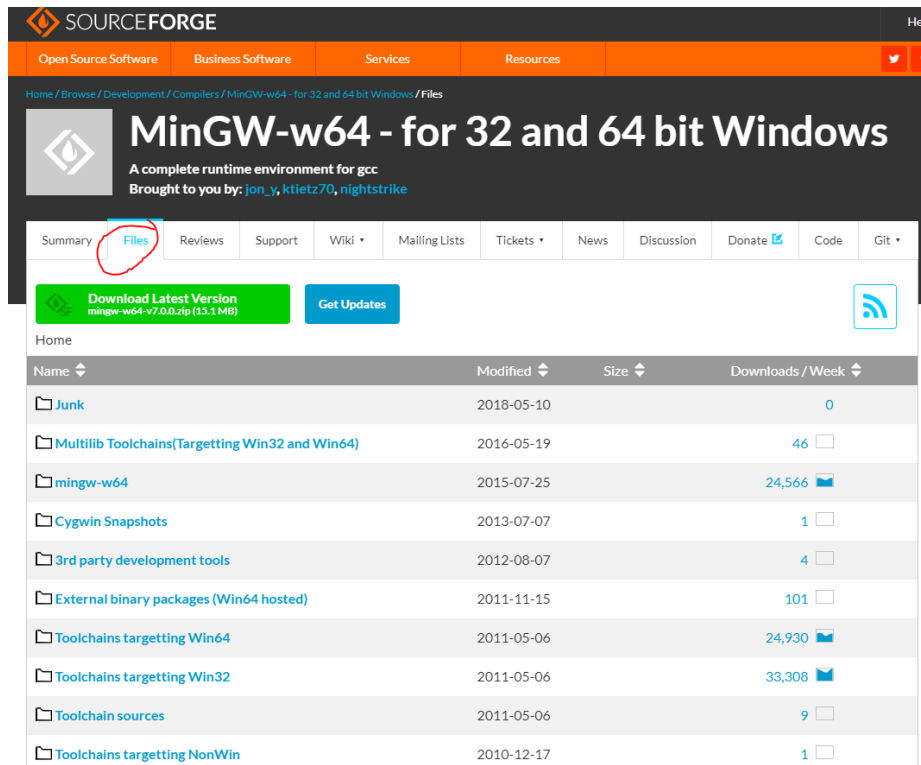
# 繳交和執行環境

- 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
  - gcc vesion: 11.3.0
  - Python version: python 3.9.13

# **GCC in windows**

# In Windows (Step 1)

- Download the compiler installer
- <https://sourceforge.net/projects/mingw-w64/>



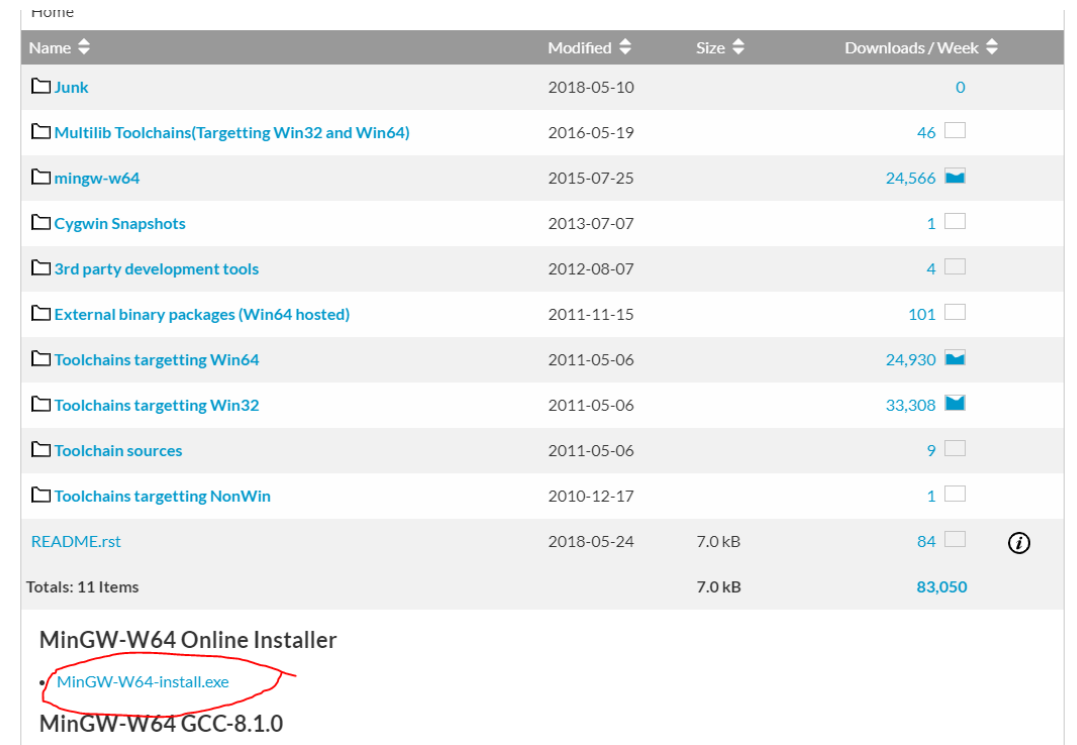
**MinGW-w64 - for 32 and 64 bit Windows**  
A complete runtime environment for gcc  
Brought to you by: jon\_y, ktietz70, nightstrike

Summary **Files** Reviews Support Wiki Mailing Lists Tickets News Discussion Donate Code Git

**Download Latest Version**  
mingw-w64-v7.0.0.zip (15.1 MB)

**Get Updates**

Name	Modified	Size	Downloads / Week
Junk	2018-05-10		0
Multilib Toolchains(Targetting Win32 and Win64)	2016-05-19		46
mingw-w64	2015-07-25		24,566
Cygwin Snapshots	2013-07-07		1
3rd party development tools	2012-08-07		4
External binary packages (Win64 hosted)	2011-11-15		101
Toolchains targeting Win64	2011-05-06		24,930
Toolchains targeting Win32	2011-05-06		33,308
Toolchain sources	2011-05-06		9
Toolchains targeting NonWin	2010-12-17		1



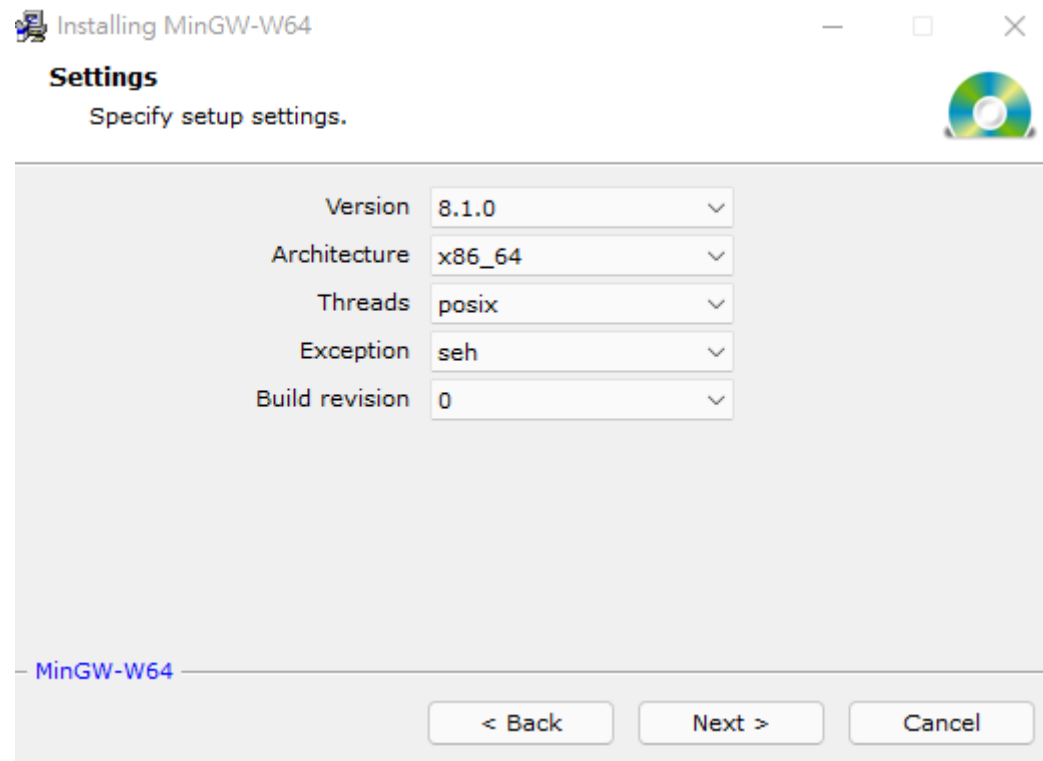
Name	Modified	Size	Downloads / Week
Junk	2018-05-10		0
Multilib Toolchains(Targetting Win32 and Win64)	2016-05-19		46
mingw-w64	2015-07-25		24,566
Cygwin Snapshots	2013-07-07		1
3rd party development tools	2012-08-07		4
External binary packages (Win64 hosted)	2011-11-15		101
Toolchains targeting Win64	2011-05-06		24,930
Toolchains targeting Win32	2011-05-06		33,308
Toolchain sources	2011-05-06		9
Toolchains targeting NonWin	2010-12-17		1
README.rst	2018-05-24	7.0 kB	84
Totals: 11 Items		7.0 kB	83,050

**MinGW-W64 Online Installer**

- MinGW-W64-install.exe

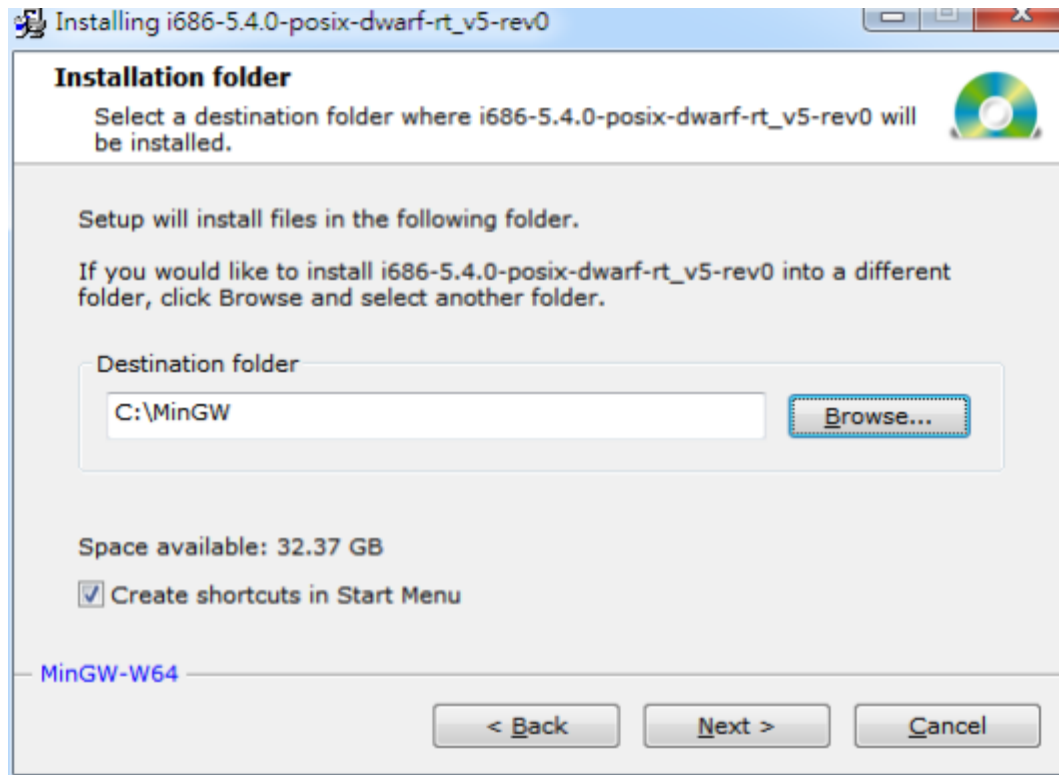
MinGW-W64 GCC-8.1.0

# In Windows (Step 2)



# In Windows (Step 3)

Install the compiler to C:\MinGW



# In Windows (Step 4)

- Computer [right button]->Properties
- - >Advanced system settings- >Advanced->Environment Variables
- -> System Variables->Path->add the instruction C:\MinGW\mingw64\bin;
- Now you can use g++ in cmd



系統

← → ↑ 控制台 > 系統及安全性 > 系統 搜尋控制台

控制台首頁

- 裝置管理員
- 遠端設定
- 系統保護
- 進階系統設定

### 檢視電腦的基本資訊

Windows 版本

Windows 10 教育版

© 2018 Microsoft Corporation. 著作權所有，並保留一切權利。

系統

處理器: Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz 3.70 GHz

已安裝記憶體 (RAM) 8.00 GB

系統類型: 64 位元作業系統, x64 型處理器

手寫筆與觸控: 此顯示器不提供手寫筆或觸控式輸入功能。

電腦名稱、網域及工作群組設定

電腦名稱: DESKTOP-QQ598IN

完整電腦名稱: DESKTOP-QQ598IN

電腦描述:

工作群組: WORKGROUP

Windows 啟用

Windows 已啟用 [閱讀 Microsoft 軟體授權條款](#)

產品識別碼: 00328-10000-00001-AA553

請參閱  
安全性與維護

系統內容

電腦名稱 硬體 進階 系統保護 遠端

您必須以系統管理員的身分登入，才能變更這裡的大部分設定。

效能

視覺效果、處理器排程、記憶體使用量和虛擬記憶體

設定(S)...

使用者設定檔

關於您登入時的桌面設定

設定(E)...

啟動及修復

系統啟動、系統失敗、及偵錯資訊

設定(T)...

環境變數(N)...

確定 取消 套用(A)

環境變數

Nacujachu 的使用者變數(U)

變數	值
OneDrive	C:\Users\Nacujachu\OneDrive
Path	C:\Users\Nacujachu\Anaconda3;C:\Users...
TEMP	C:\Users\Nacujachu\AppData\Local\Temp
TMP	C:\Users\Nacujachu\AppData\Local\Temp

新增(N)... 編輯(E)... 刪除(D)

系統變數(S)

變數	值
OS	Windows_NT
Path	C:\Program Files\NVIDIA GPU Computing...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;...
PROCESSOR_AR...	AMD64

新增(W)... 編輯(I)... 刪除(L)

確定 取消

# For Your Reference

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