# Data Science
# Coding Assignment #2
# Frequent Patterns

Submission Prejudge Deadline:

2018/10/17 23:59

Submit to e3

Submission Final Deadline:

2018/10/22 23:59

Submit to e3

# 目標

- 給定transaction dataset 和 min support，列出frequent patterns
- 實作任一種演算法 Eg. Apriori, Fpgrowth, ECLAT
- **正確**和**快**!!!!!!

# Requirements

- Implement with C, C++, python3.6
- Strictly follow input/output formats
- Do not copy/paste others' codes
  - You can refer to the codes on GitHub or anywhere else
  - But you need to write your own code

# Input File Format

- Item以數字編號表示(0~999)
- 一行為一筆transaction(最多500,000筆)
- 每筆transaction的item數量不一致(至少1項，最多100項)
- 每筆transaction已按item編號排序
- Sample Input

```
0,1,2,5
0,2,
0,2,3,5
1,2,4
1,3
0,2,3,5
1,3
1,2,3,5
1,2,3
```

逗號分隔(前後無空格)，每行最後無空格

# Load Input Sample C++ code

```
#include <iostream>
#include <cstdio>
#include <sstream>
....
int main(void)
{
    //redirect stdin to 1.in file
    freopen("testcase.in","r", stdin);
    vector<vector<int>> transactions;
    string line;
    //read until EOF
    while(!getline(cin, line).eof()){
            vector<int> arr;
            istringstream ssline(line);
            string number;
            while(getline(ssline, number, ','))
                arr.push_back(atoi(number.c_str()));
            transactions.push_back(arr);
    }
    return 0;
}
```

# Output File Format

- 一行為一組frequent pattern
- 輸出順序
  - item數量少的pattern在前，數量多的在後
  - item數量相等的patterns，item編號起始小者在前...以此類推
  - Pattern內item編號由小到大
- Sample output(min_support=0.4)

```
0:0.4444
1:0.6667
2:0.7778
3:0.6667
5:0.4444
0,2:0.4444
1,2:0.4444
1,3:0.4444
2,3:0.4444
2,5:0.4444
```

- $Support(A) = \frac{\text{A數量}}{transactions\text{數量}}$
- 一行為一組Frequent pattern與Support
- Frequent pattern與Support間以小寫冒號分隔
  (Frequent pattern:support)
- Frequent pattern中，item間以逗號分隔
- Support輸出請四捨五入至小數點第四位
- 逗號、冒號前後皆無空白

# Notes

- 內建的round function"或許"跟你想的不一樣
  - 以python為例，如下:
    ```
    >>> a = 0.33575
    >>> print(round(a, 4))
    0.3357
    >>> b = 0.12345
    >>> print(round(b, 4))
    0.1235
    ```
  - 因此"直接使用" round function會有誤。
  - p.s.使用C++的同學也要注意

# Deadline

- Prejudge Deadline(10/17)
  - 助教會幫大家預測試3筆測資(最終6筆測資的其中3筆)
  - 10/19公布3筆測資的分數和執行時間

- Final Deadline(10/22)
  - 最終成績

# 評分方式與配分(Final Deadline)

- **評分方式**
  - 助教執行上傳的程式
  - 並測試六組測資(特定min support)
- **配分 (total 100%，正確性、速度比拚)**

| | 正確性 | 速度比拚～～不正確就沒得比了!<br>全部比賽人數平均成四個速度等級 | Timeout |
|---|---|---|---|
| Testcase 1 | 15% | 不比 | |
| Testcase 2 | 15% | 不比 | |
| Testcase 3 | 10% | 9%(按等級分別得2%, 4%, 6%, 9%) | 130s |
| Testcase 4 | 10% | 9%(按等級分別得2%, 4%, 6%, 9%) | 275s |
| Testcase 5 | 10% | 6%(按等級分別得1%, 2%, 4%, 6%) | 950s |
| Testcase 6 | 10% | 6%(按等級分別得1%, 2%, 4%, 6%) | 2170s |

# 測試環境

- OS : Windows10
- CPU : 4核心, 8執行緒, 3.2GHz(可以用使用multiple processing寫法)
- RAM : DDR4 16G
- Compiler : g++ 5.4.0 (c++11)
- Disk : 1TB

# 測試方法

- min support : 0.01～0.5
- C++ code
  - 助教會用提供的makefile去compile程式
  - 以下方方法執行:
    [執行檔] [min_support] [inputFile(測資)] [outputFile]
  - 務必確定程式可以依此方法執行

```
E:\資料科學助教\cpp_version>main.exe 0.1 testcase1.txt ans1.txt
```

- Python code
  - 執行 : python [py檔] [min_support] [inputFile(測資)] [outputFile]
- outputFile 再與答案比對

# 作業繳交格式與命名

- 上傳
  - Prejudge Deadline : “學號_HW2_prejudge.zip“
  - Final Deadline: “學號_HW2_final.zip”
- 資料夾內含(c++)
  - cpp主程式，請統一命名為 “main.cpp”
  - 其餘你所需要的cpp, header檔
  - readme(如果有特殊header檔，請在此詳述使用原因)
- 資料夾內含(python)
  - python主程式，請統一命名為“main.py”
  - 其餘你所需要的py檔
  - readme
    - 列出使用的library
    - 其中若有非資料處理等常用的library，請在此詳述使用原因

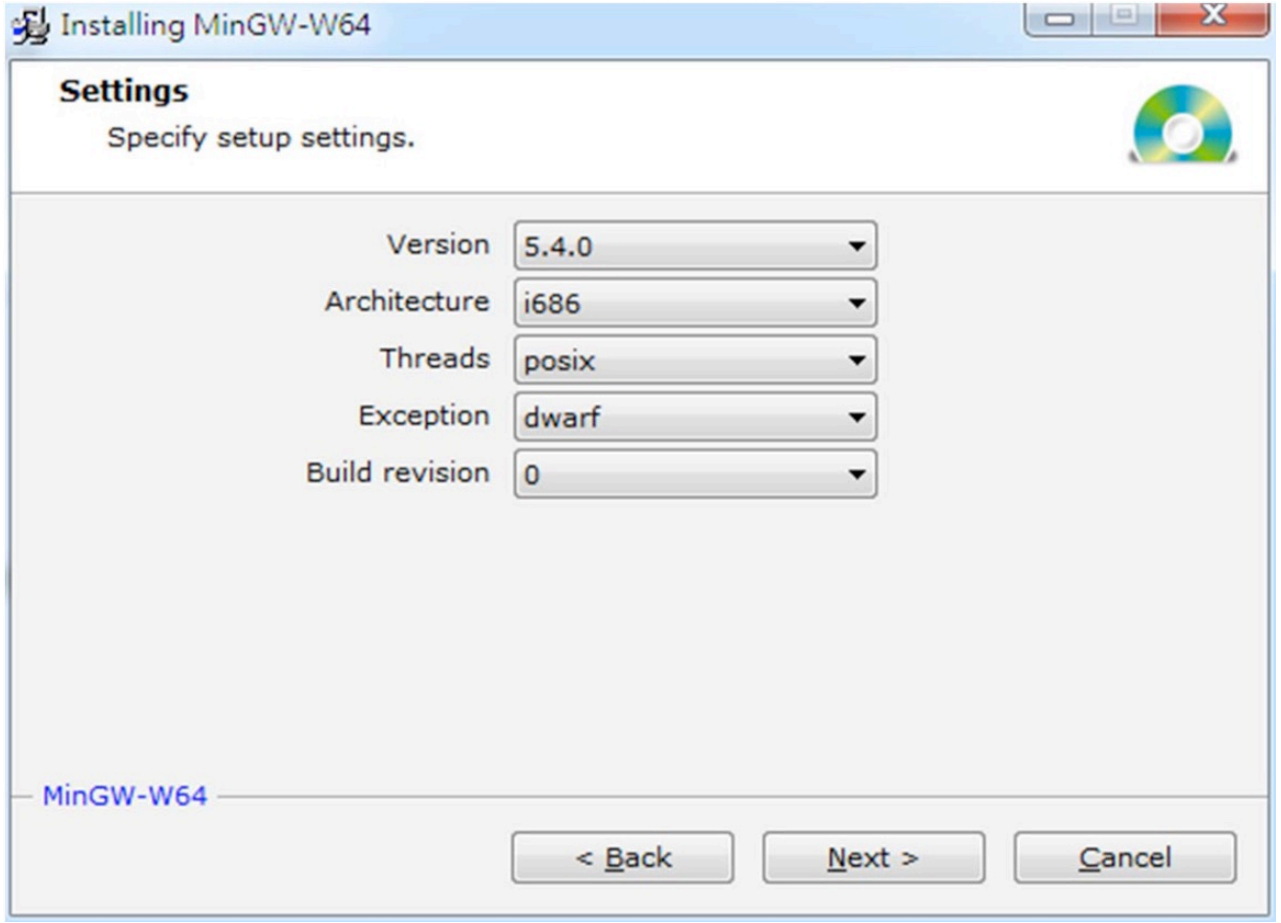
# How to use Makefile

# In Windows (Step 1)

- Download the compiler installer(both x86 and x86_64)
  - https://sourceforge.net/projects/mingw-w64/

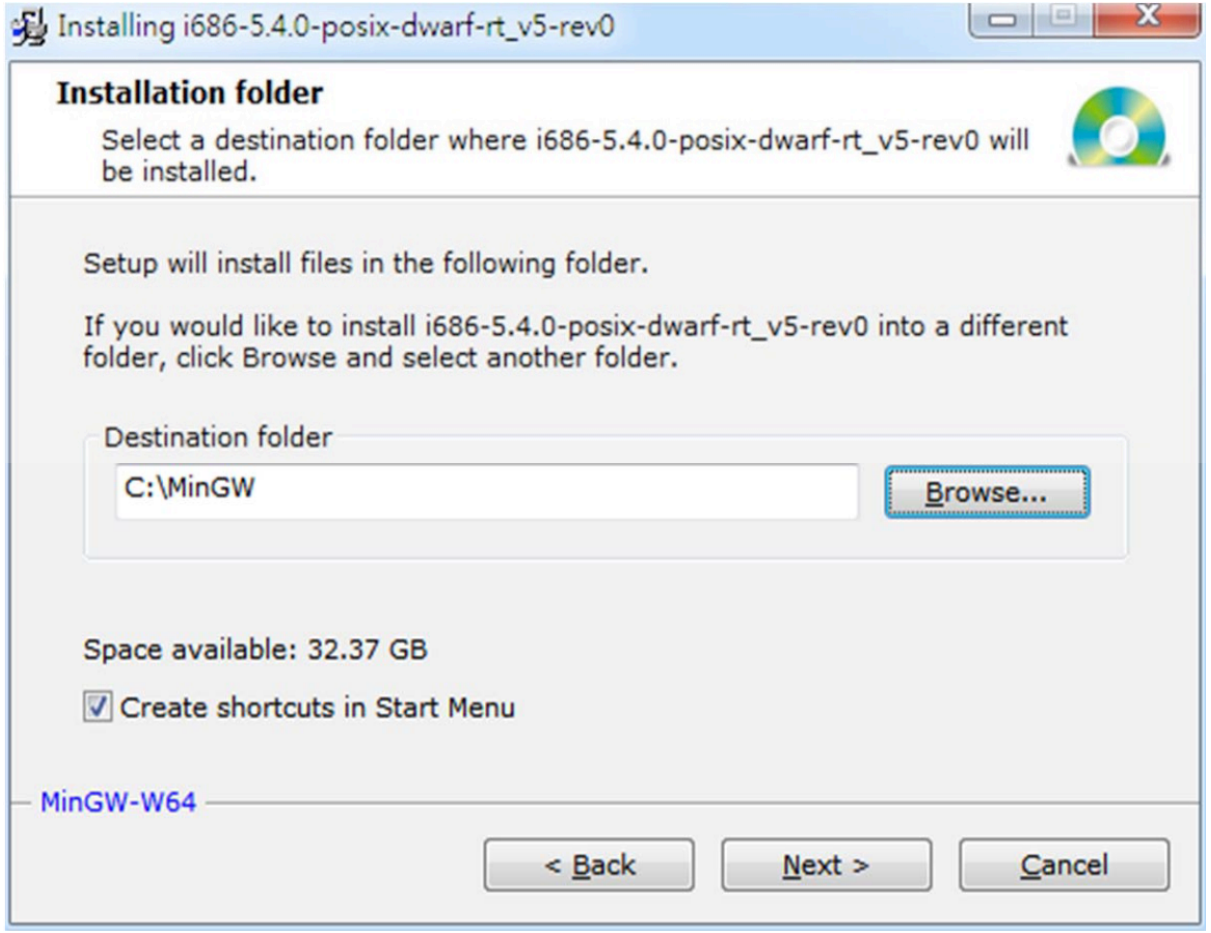
# In Windows (Step 2)

- Select gcc version 5.4.0 and i686 architecture



# In Windows (Step 3)

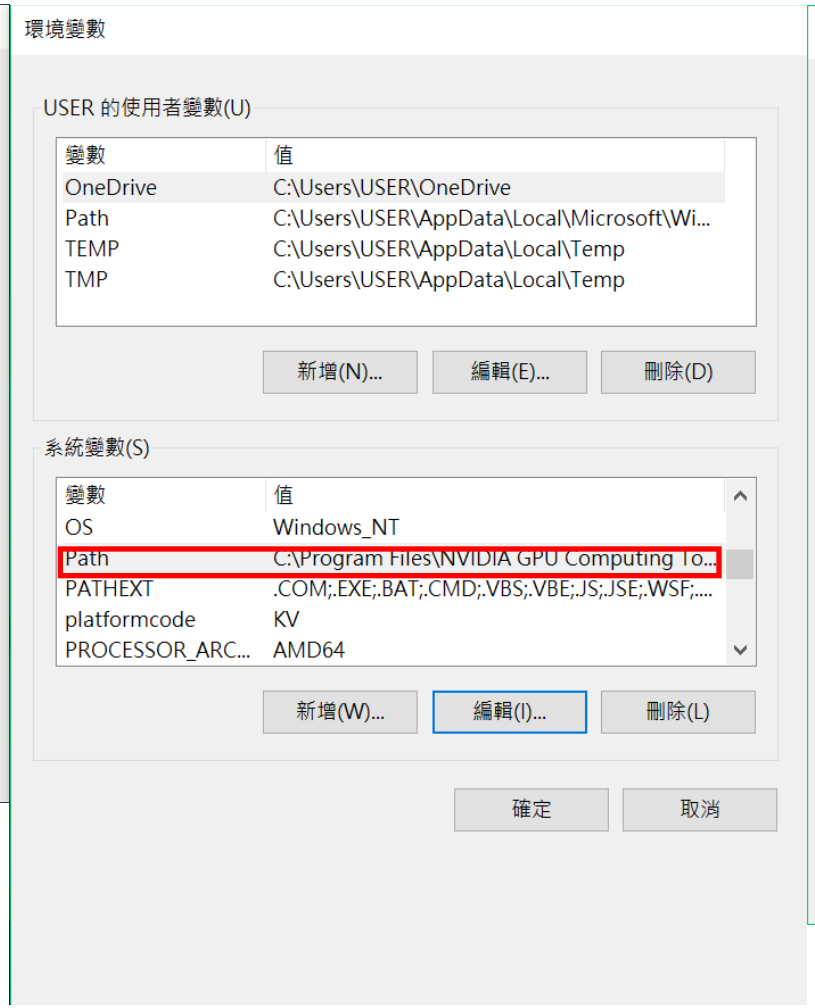
- Install the compiler to C:\MinGW

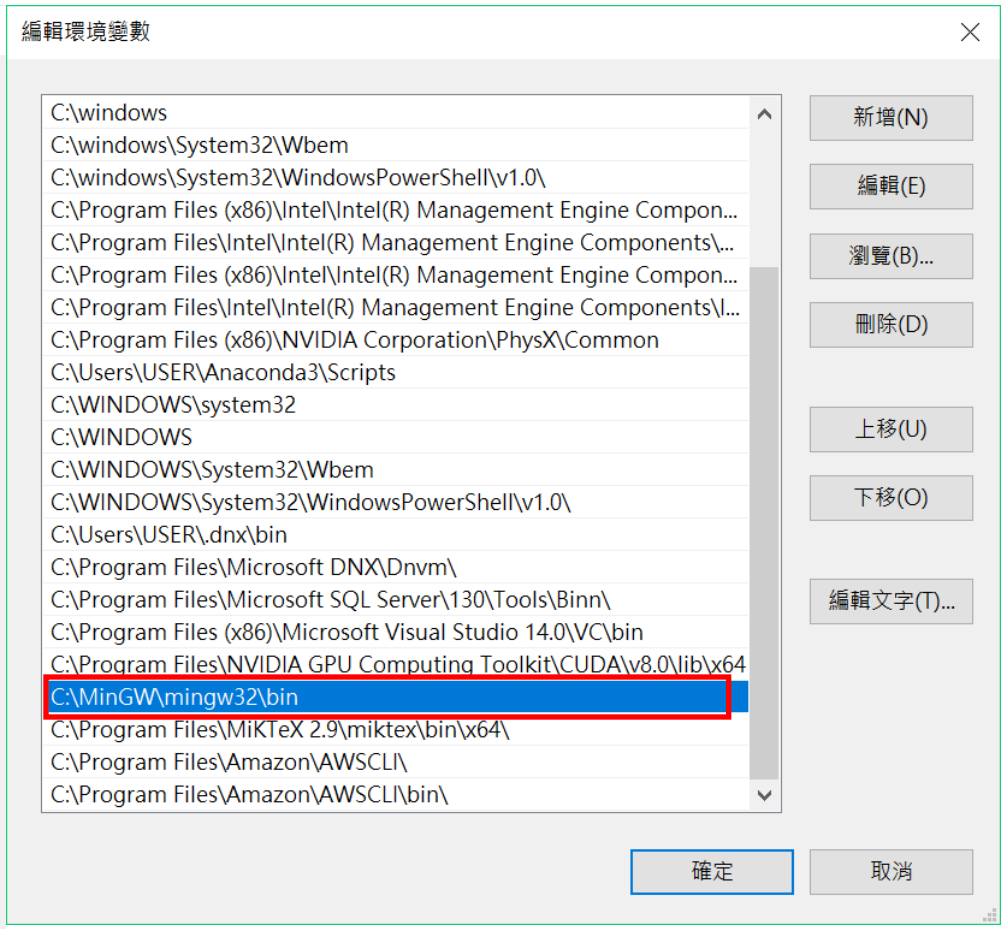


# In Windows (Step 4)

- Computer [right button]->Properties->Advanced system settings->Advanced->Environment Variables-> System Variables->Path->add the instruction C:\MinGW\mingw32\bin;
- Now you can use g++ in cmd

# In Windows (Step 4)

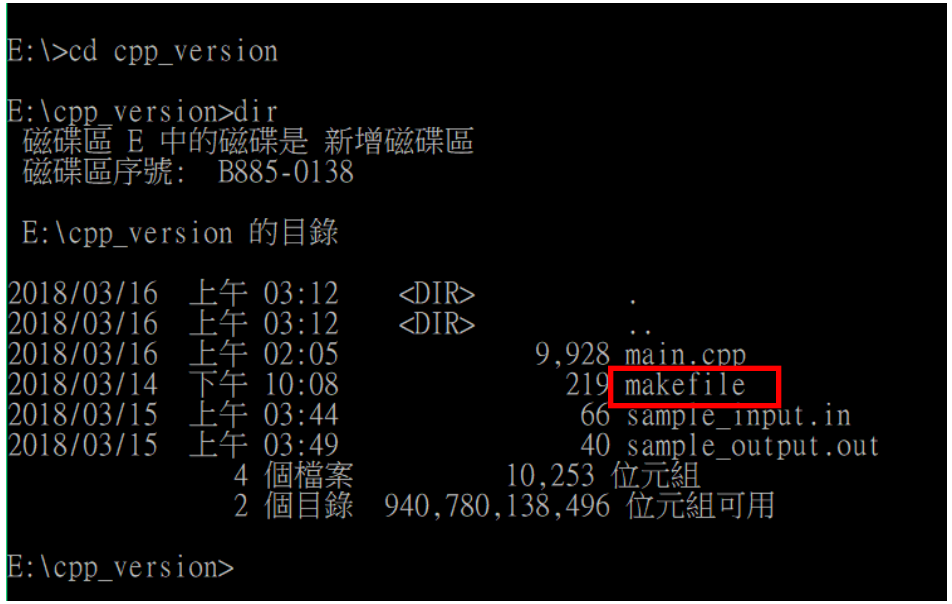



# An Example

- Start Menu->search “cmd”->enter “gcc -v” (If it is successful, the cmd would show gcc verision 5.4.0)

# In Windows(makefile)

- Makefile will help you compile the program




- Enter "mingw32-make (-f makefile)" and it will create a executable file.

# Questions?