# Mining Frequent Patterns With Multiple Minimum Supports in Transactional Databases

## What is frequent pattern mining with multiple minimum support?

Frequent pattern mining with multiple minimum support aims to discover all interesting patterns in a transactional database that satisfies multiple minimum support specified by the user.

Reference: Bing Liu, Wynne Hsu, and Yiming Ma. 1999. Mining association rules with multiple minimum supports. In Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '99). Association for Computing Machinery, New York, NY, USA, 337–341.
https://doi.org/10.1145/312129.312274

## What is a transactional database?

A transactional database is a collection of transactions, where each transaction contains a transaction-identifier and a set of items.
A hypothetical transactional database containing the items ***a, b, c, d, e, f, and g*** as shown below.

| tid | Transactions |
|---|---|
| 1 | a b c g |
| 2 | b c d e |
| 3 | a b c d |
| 4 | a c d f |
| 5 | a b c d g |
| 6 | c d e f |
| 7 | a b c d |
| 8 | a e f |
| 9 | a b c d |
| 10 | b c d e |

**Note:** Duplicate items must not exist in a transaction.

## What is acceptable format of a transactional databases in PAMI?

Each row in a transactional database must contain only items. The frequent pattern mining algorithms in PAMI implicitly assume the row number of a transaction as its transactional-identifier to reduce storage and processing costs. A sample transactional database, say sampleInputFile.txt, is provided below.

a b c g
b c d e
a b c d
a c d f
a b c d g
c d e f
a b c d
a e f
a b c d
b c d e

## Understanding the statisctics of database

To understand about the database. The below code will give the detail about the transactional database.

- Total number of transactions (Database size)
- Total number of unique items in database
- Minimum lenth of transaction that existed in database
- Average length of all transactions that exists in database
- Maximum length of transaction that existed in database
- Standard deviation of transaction length
- Variance in transaction length
- Sparsity of database

The below sample code prints the statistical details of a database.

In [ ]:
```
import PAMI.extras.dbStats.transactionalDatabaseStats as stats
obj = stats.transactionalDatabaseStats('sampleInputFile.txt', ' ')
obj.run()
obj.printStats()
```

# What is a multiple minimum support file?

A multiple minimum support file contains a set of times and minimum support values. A hypothetical multiple minSup database containing the items ***a, b, c, d, e, f, and g*** as shown below.

a 4
b 4
c 3
d 4
e 4
f 5
g 5

# What are the input parameters?

The input parameters to a frequent pattern mining algorithm are:

- **Transactional database**
  Acceptable formats:

  > - String : E.g., 'transactionalDatabase.txt'
  > - URL : E.g., https://u-aizu.ac.jp/~udayrage/datasets/transactionalDatabases/transactional_T10
  > - DataFrame with the header titled 'Transactions'

- **multiple minSup value database**
  Acceptable formats:

  > - String : E.g., 'minSupDatabase.txt'
  > - URL : E.g., https://u-aizu.ac.jp/~udayrage/datasets/transactionalDatabases/transactional_T10
  > - DataFrame with the header titled 'item' and 'MIS'

- **seperator**
  default seperator is '\t' (tab space)

# How to store the output of a frequent pattern mining with multiple minimum support algorithm?

The patterns discovered by a frequent pattern mining with multiple minimum support algorithm can be saved into a file or a data frame.

- Download the PAMI source code from github.
- Unzip the PAMI source code folder and enter into multiple minimum support frequent patterns folder.
- Enter into multipleminimumSupportbasedFrequentPattern folder
- Enter into a specific folder of your choice and execute the following command on terminal.

**syntax:** python3 algorithmName.py `<path to the input file>` `<path to the output file>` `<path to the MIS file` `<seperator>`

**Example:** python3 `CFPGrowth.py` `inputFile.txt` `outputFile.txt` `MIS.txt` `' '`

# How to execute a frequent pattern mining with multiple minimum supports algorithm in a Jupyter Notebook?

- Install the PAMI package from the PYPI repository by executing the following command: **pip3 install PAMI**
- Run the below sample code by making necessary changes

In [ ]:
```
import PAMI.multipleMinimumSupportBasedFrequentPattern.CFPGrowth as alg

iFile = 'sampleInputFile.txt'  #specify the input transactional database
mFile = 'MIS.txt'
seperator = ' ' #specify the seperator. Default seperator is tab space. <
oFile = 'frequentPatterns.txt'   #specify the output file name<br>

obj = alg.CFPGrowth(iFile, mFile, seperator) #initialize the algorithm <b
obj.startMine()                         #start the mining process <br>
obj.savePatterns(oFile)                 #store the patterns in file <br>
df = obj.getPatternsAsDataFrame()       #Get the patterns discovered into a
obj.printStats()                        #Print the statistics of mining pro
```

The frequentPatterns.txt file contains the following patterns (*format:* pattern:support):!cat frequentPatterns.txt

In [4]:
```
!cat frequentPatterns.txt
```

```
e :4
e d c :3
e c :3
b :7
b a :5
b a d :4
b a d c :4
b a c :5
b d :6
b d c :6
b c :7
a :7
a d :5
a d c :5
a c :6
d :8
d c :8
c :9
```

The dataframe containing the patterns is shown below:

In [5]:
```
df
```

Out[5]:

| | Patterns | Support |
|---|---|---|
| **0** | e | 4 |
| **1** | e d c | 3 |
| **2** | e c | 3 |
| **3** | b | 7 |
| **4** | b a | 5 |
| **5** | b a d | 4 |
| **6** | b a d c | 4 |
| **7** | b a c | 5 |
| **8** | b d | 6 |
| **9** | b d c | 6 |
| **10** | b c | 7 |
| **11** | a | 7 |
| **12** | a d | 5 |
| **13** | a d c | 5 |
| **14** | a c | 6 |
| **15** | d | 8 |
| **16** | d c | 8 |
| **17** | c | 9 |