Applying gSpan algorithm on biological dataset using SPMF library in Python

TASK :2

# Description:

**SPMF**is an **open-source software**and**data mining library**written in **Java**, specialized in **pattern** mining .

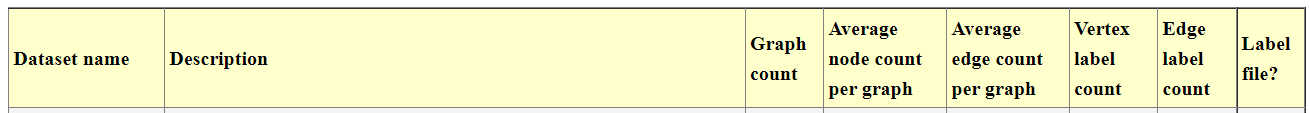
## If you want to use SPMF from other languages besides Java, then SPMF provide “Wrappers”.

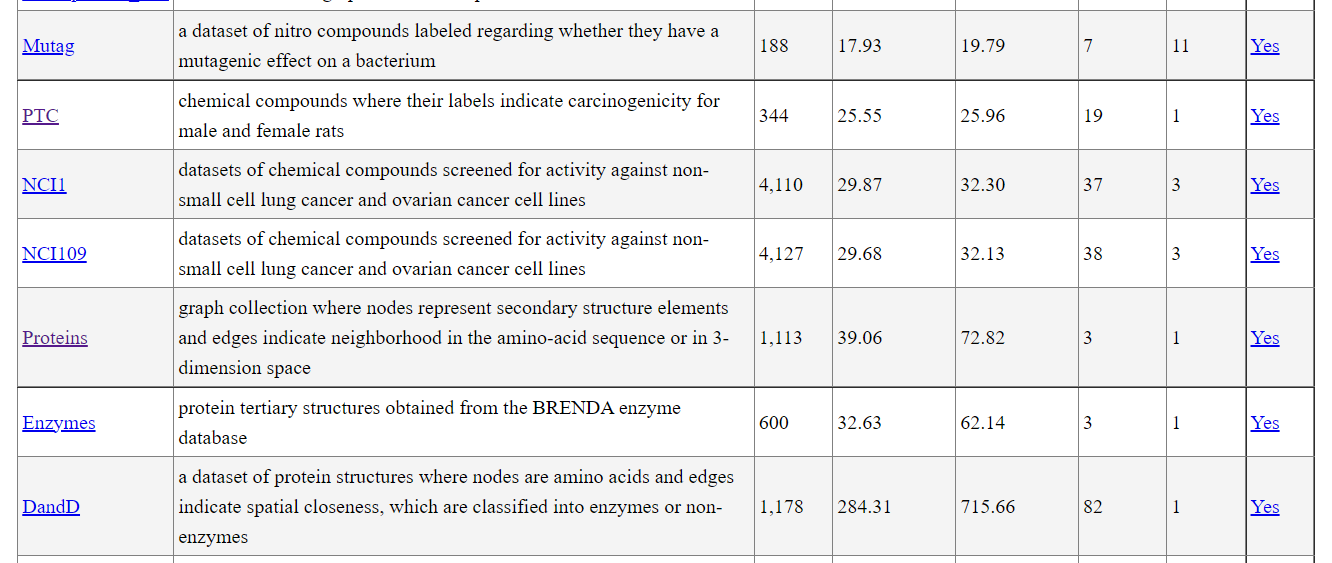
### Here I used . Py-SPMF wrapper for Python.which made by “Lorenz Leitner”.

The algorithm takes a graph database as input. Then the frequent subgraph mining algorithm will enumerate as output all frequent subgraphs.

A **frequent subgraph** is a subgraph that appears in at least *minsup* percents of the graphs from the graph database

# Graph database:





the following graph database name is “[Enzymes](https://www.philippe-fournier-viger.com/spmf/datasets/dang/enzymes_graph.txt)” ,a protein tertiary structures obtained from the BRENDA enzyme database three graphs, which is provided as text file by the SPMF software .

The algorithm takes a **graph database as input.**

Then, the frequent subgraph mining algorithm will enumerate as output all frequent subgraphs.

A **frequent subgraph** is a subgraph that appears in at least *minsup* percents of the graphs from the graph database

Description: A picture containing table

Description automatically generated

• t # 0 This is the first line of a graph. It indicates that this is the N-th graph in the file.

• v M L This line defines the M-th vertex of the current graph, which has a label L

• e P Q L This line defines an edge, which connects the P-th vertex with the Q-th vertex. This edge has the label L

Description: A picture containing table

Description automatically generated

# Goal :

We want to applying the gSpan algorithm, to get the set of all subgraphs appearing from the graphed Enzyme dataset.

# Output file format

The output file format is defined as follows. It is a text file, listing all the frequent subgraphs found in the input graph database. A frequent subgraph is defined by a few lines of text that follow the following format:

**t # N \* Z** This is the first line of a subgraph. It indicates that this is the N-th subgraph in the file and that its support is Z.

**v M L** This line defines the M-th vertex of the current subgraph, which has a label L

**e P Q L** This line defines an edge, which connects the P-th vertex with the Q-th vertex. This edge has the label L

**x X1 X2** ... This lines lists the identifiers of all the graphs X1, X2 ... that contains the current subgraph.

# First : install spmf.jar

# To install the wrapper, it requires downloading the spmf.jar Which contain the java code library of spmf.

Then un-zip the file in the jupyter , now we can import the SPMF library in python jupyter environment .

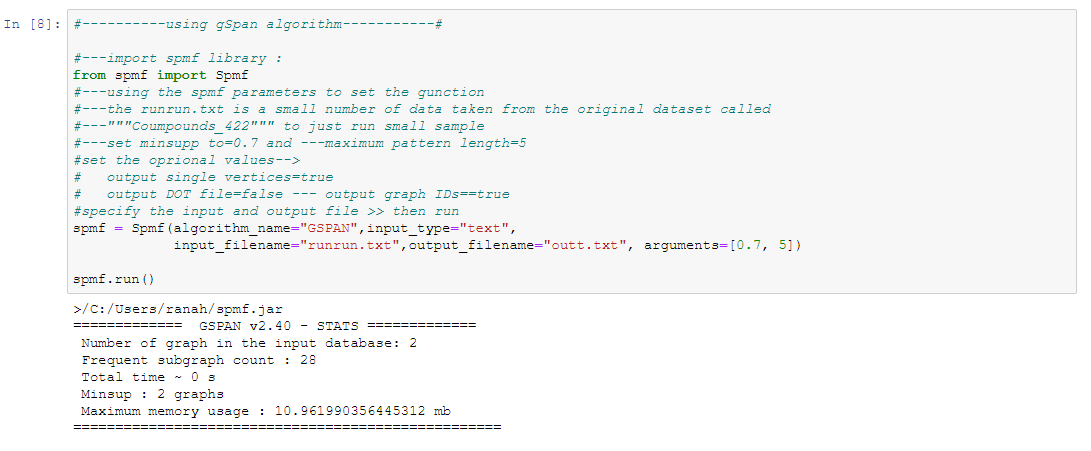
# Example:

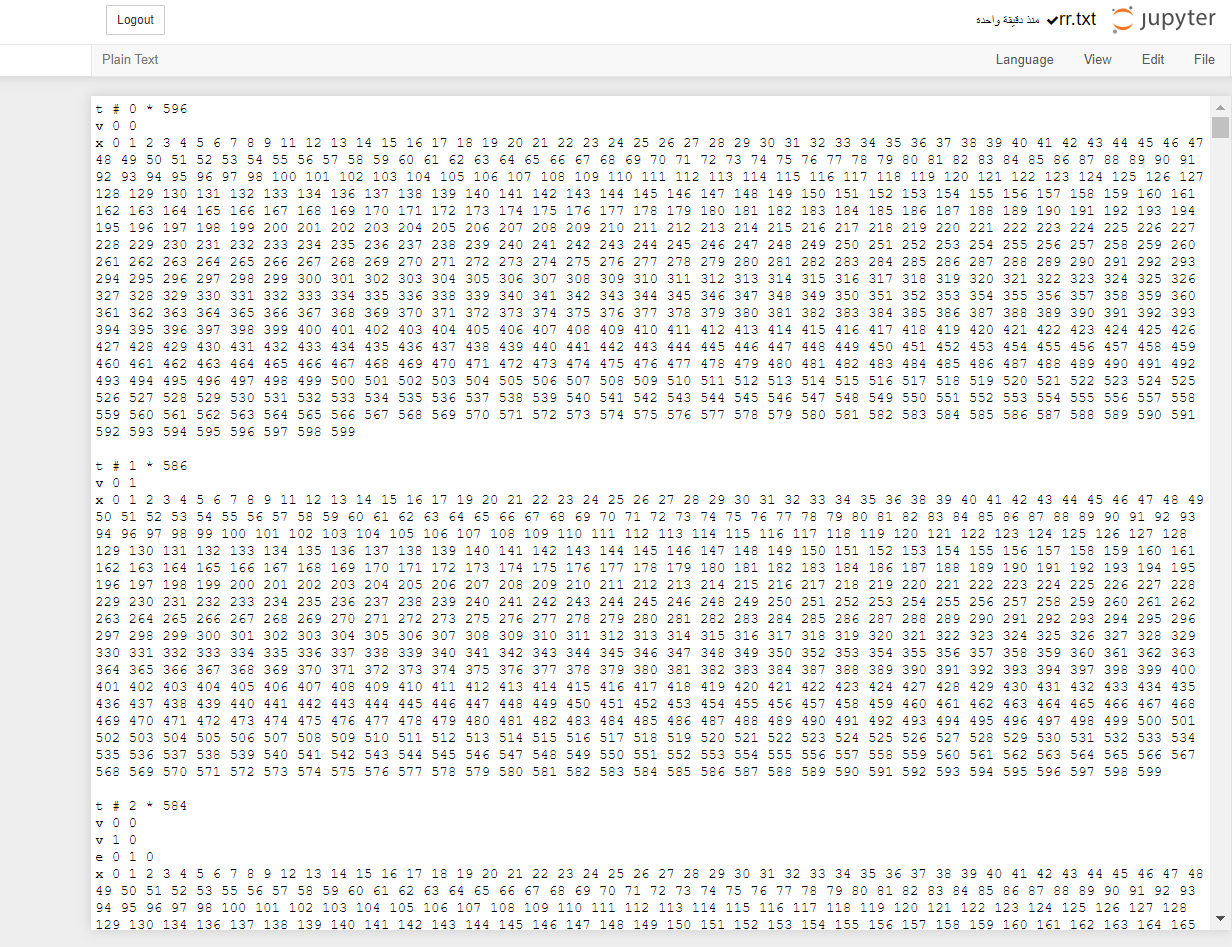
* Input file: biological dataset “Enzyme.txt”
* Using minsupp = 0.7
* maximum pattern length =5

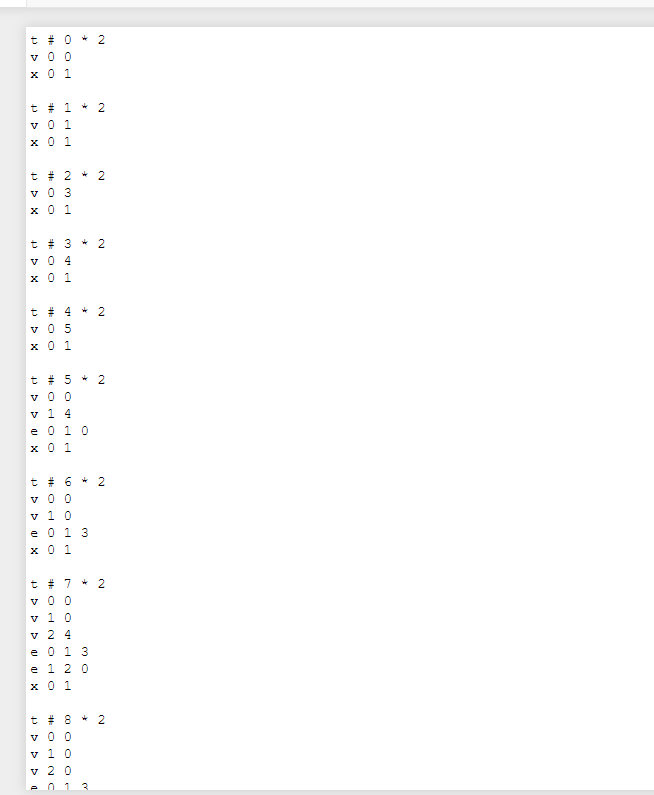
# Jupyter file run :



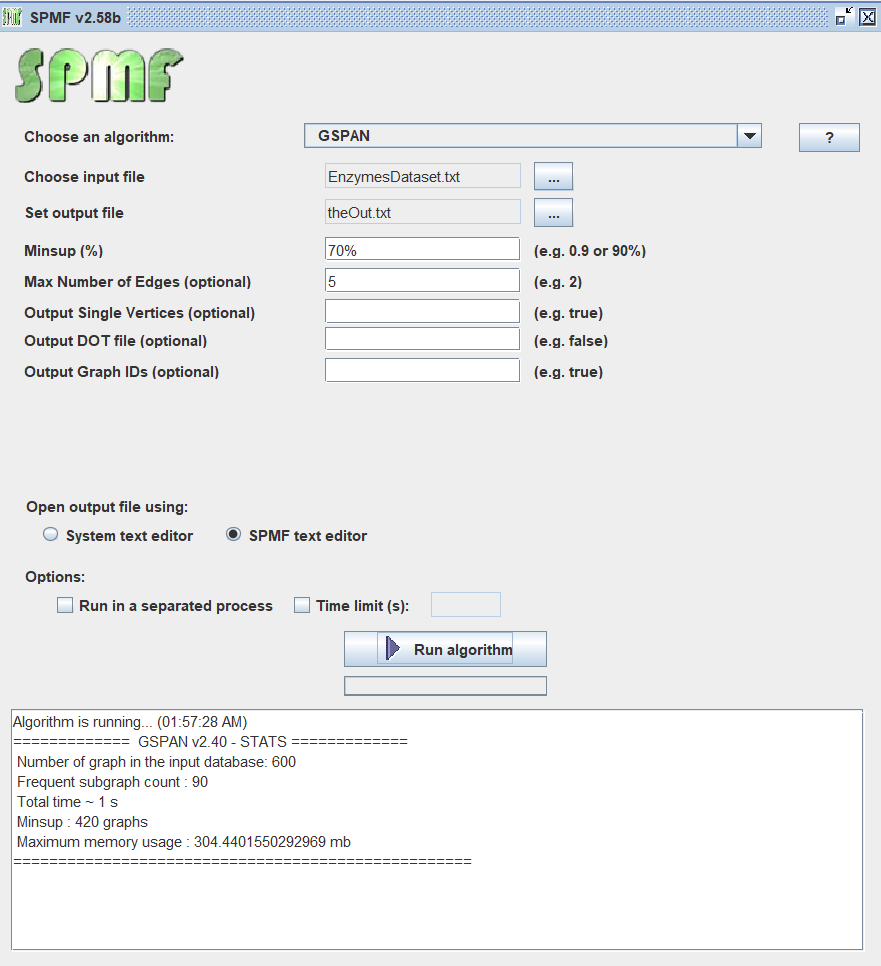
Try another dataset called [Chemical\_340](https://www.philippe-fournier-viger.com/spmf/datasets/Chemical_340.txt).text a database of 340 graphs about chemistry

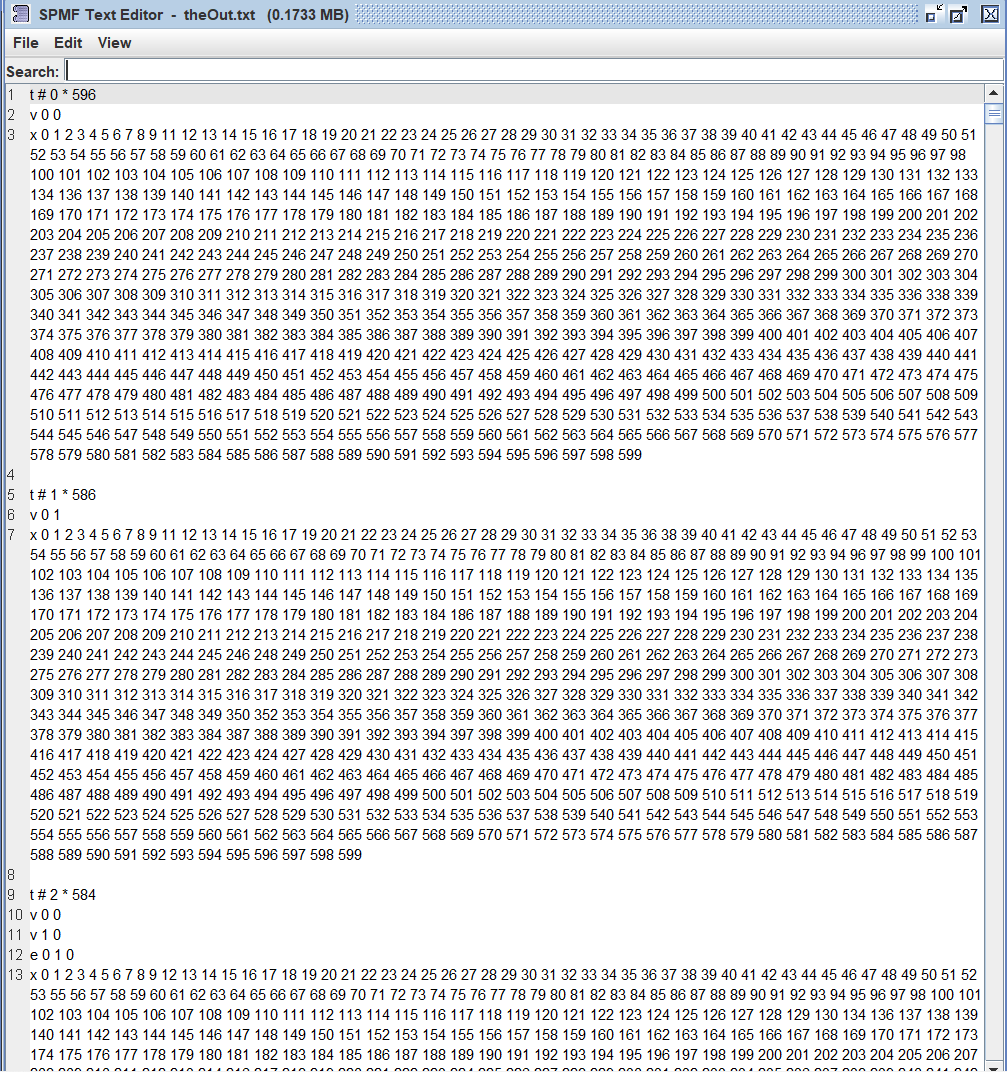


**The rr.txt file(output file):**

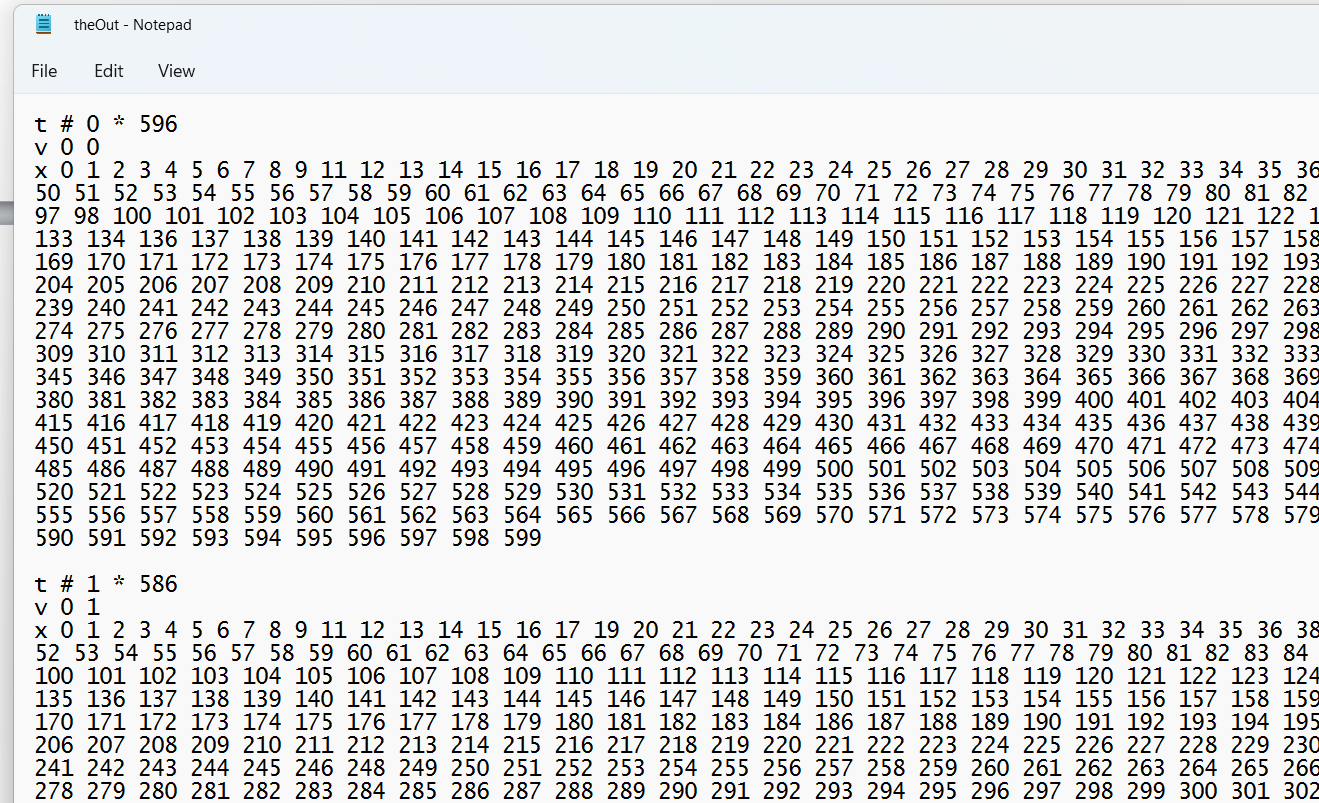
**The outt.txt file(output file)**

# Also run SPMF software to make sure :

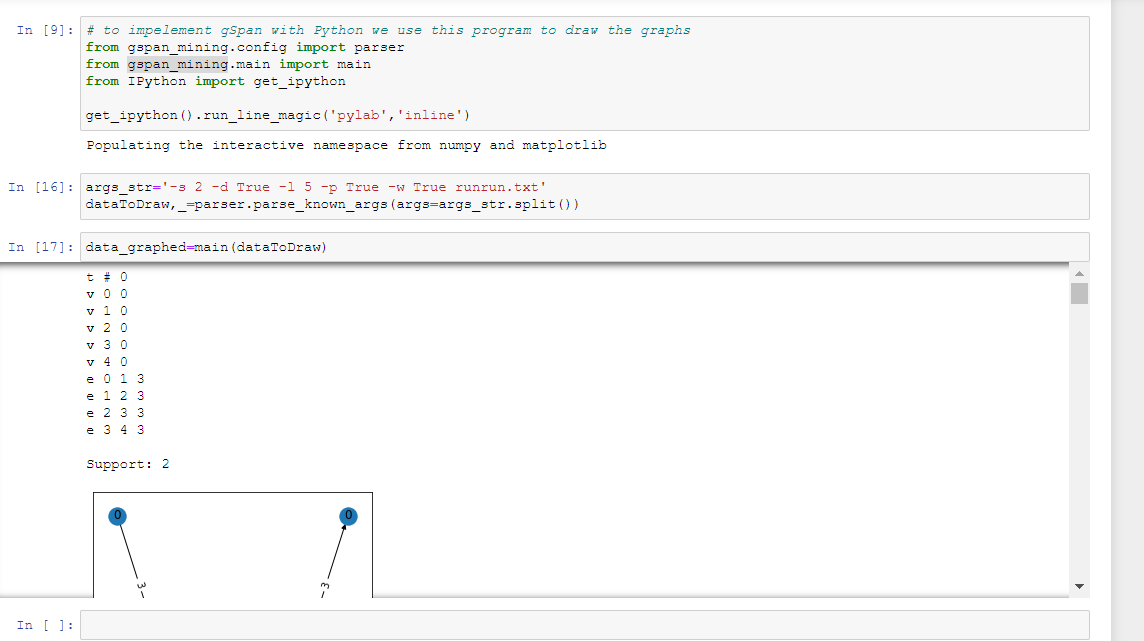
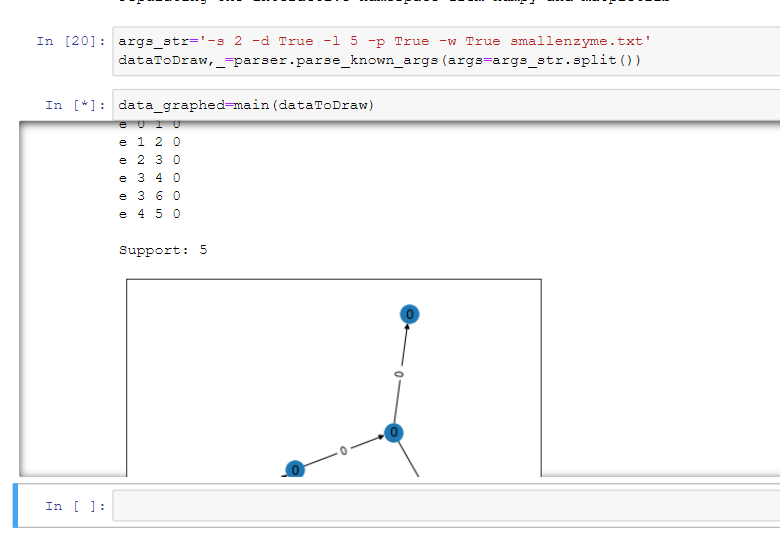
* choose gspan algorithim,input and output files, then run :



# Output file of SPMF:



Now apply the graph for the second dataset:

REFRENCE CODE:

Guithub :