Advanced Tutorial on Implementing FSPGrowth Algorithm

In this tutorial, we explain how the FSPGrowth algorithm can be implemented by varying the minimum support values

Step 1: Import the FSPGrowth algorithm and pandas data frame

In [1]:

```
from PAMI.frequentSpatialPattern.basic import FSPGrowth  as alg
import pandas as pd
```

Step 2: Specify the following input parameters

In [2]:

```
inputFile = 'transactional_T10I4D100K.csv'
seperator='¥t'
minimumSupportCountList = [100, 150, 200, 250, 300]
#minimumSupport can also specified between 0 to 1. E.g., minSupList = [0.005, 0.006,
neighborFile='T10_utility_neighbour.txt'
result = pd.DataFrame(columns=['algorithm', 'minSup', 'patterns', 'runtime', 'memory
#initialize a data frame to store the results of FSPGrowth algorithm
```

Step 3: Execute the FSPGrowth algorithm using a for loop

In [3]:

```
algorithm = 'FSPGrowth'  #specify the algorithm name
for minSupCount in minimumSupportCountList:
    obj = alg.FSPGrowth('transactional_T10I4D100K.csv', minSup=minSupCount,nFile=ne
    obj.startMine()
    #store the results in the data frame
    result.loc[result.shape[0]] = [algorithm, minSupCount, len(obj.getPatterns()), c
```

```
Frequent Spatial Patterns successfully generated using FSPGrowth
Frequent Spatial Patterns successfully generated using FSPGrowth
Frequent Spatial Patterns successfully generated using FSPGrowth
Frequent Spatial Patterns successfully generated using FSPGrowth
Frequent Spatial Patterns successfully generated using FSPGrowth
```

In [4]:

```
print(result)
```

```
   algorithm  minSup  patterns    runtime     memory
0  FSPGrowth     100      4603  35.265882  606318592
1  FSPGrowth     150      2994  35.105762  607047680
2  FSPGrowth     200      2177  35.955468  607338496
3  FSPGrowth     250      1406  34.050695  607256576
4  FSPGrowth     300       950  32.790241  607064064
```

Step 5: Visualizing the results

Step 5.1 Importing the plot library

In [5]:

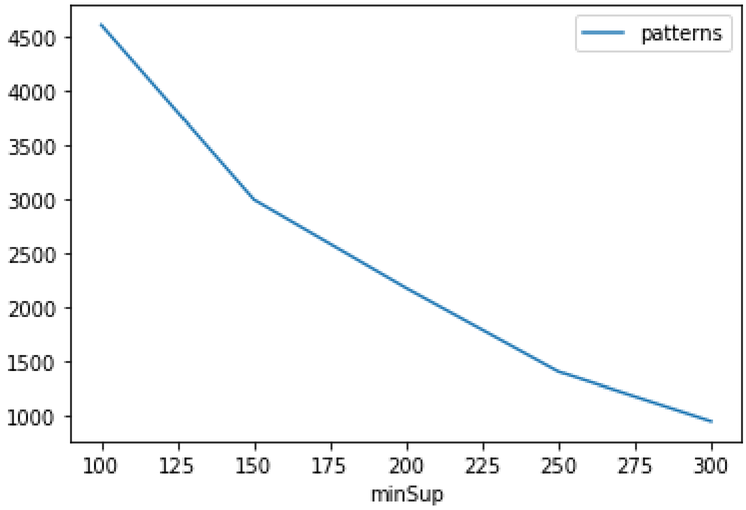
```
from PAMI.extras.graph import plotLineGraphsFromDataFrame as plt
```

Step 5.2. Plotting the number of patterns

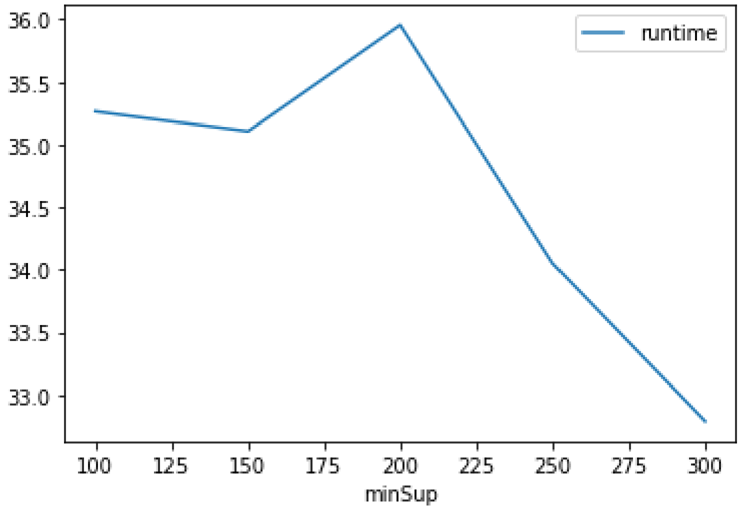
In [6]:

```
ab = plt.plotGraphsFromDataFrame(result)
```

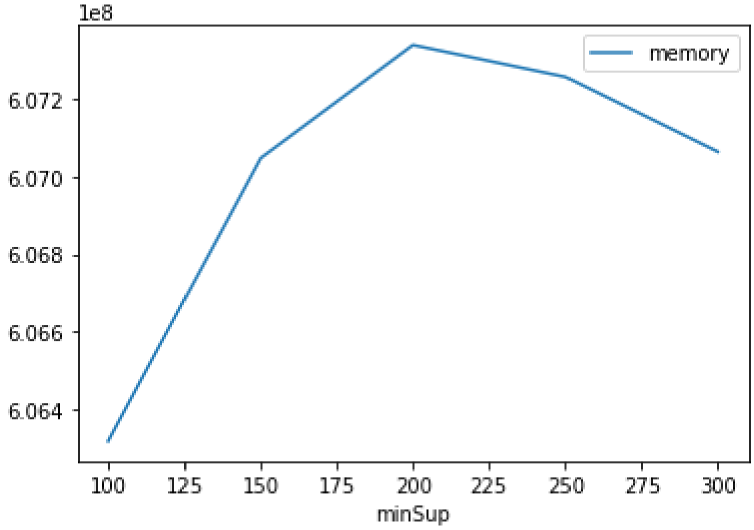
```
ab.plotGraphsFromDataFrame() #drawPlots()
```



```
Graph for No Of Patterns is successfully generated!
```



```
Graph for Runtime taken is successfully generated!
```



```
Graph for memory consumption is successfully generated!
```

Step 6: Saving the results as latex files

In [7]:

```
from PAMI.extras.graph import generateLatexFileFromDataFrame as gdf
gdf.generateLatexCode(result)
```

```
Latex files generated successfully
```