Homework 1. Frequent itemset

Double Click here to edit this cell

- Name: 조성현
- Student ID:201803430
- Submission date: 22.03.23

Remark. Do not import numpy, pandas, sklearn, or any module implementing the solution directly

Frequent itemset

- ***Support*** is an indication of how frequently the itemset X appears in the dataset T.
- The support of X with respect to T is defined as the proportion of transactions t in the dataset which contains the itemset X.

$$\text{supp}(X) = \frac{|\{t \in T; X \subseteq t\}|}{|T|}$$

- Frequent itemset is an itemset whose support $\geq$ ***min_sup***.

Data set

- Each line in the following can be imagined as a market basket, which contains items you buy.

In [1]:

```
# DO NOT EDIT THIS CELL
data_str = 'apple,beer,rice,chicken\n'
data_str += 'apple,beer,rice\n'
data_str += 'apple,beer\n'
data_str += 'apple,mango\n'
data_str += 'milk,beer,rice,chicken\n'
data_str += 'milk,beer,rice\n'
data_str += 'milk,beer\n'
data_str += 'milk,mango'
```

Problem 1 (2 pts)

- Define a function ***gen_record*** generating a list of items each ***next***.
- It must be a generator.
- Use ***yield*** instead of ***return***

In [2]:

```
# YOUR CODE MUST BE HERE
def gen_record(data_str):
    data_str = data_str.split('\n')
    for words in data_str:
        basket = sorted(words.split(','))
        yield basket
```

In [3]:

```
# DO NOT EDIT THIS CELL
test = gen_record(data_str)
print(sorted(next(test)))
```

```
['apple', 'beer', 'chicken', 'rice']
```

Your output must be:

```
['apple', 'beer', 'chicken', 'rice']
```

In [4]:

```
# DO NOT EDIT THIS CELL
print(sorted(next(test)))
```

```
['apple', 'beer', 'rice']
```

Your output must be:

```
['apple', 'beer', 'rice']
```

Problem 2 (10 pts)

- Define a function ***gen_frequent_1_itemset*** generating 1-itemset.
- It must be a generator.
- We want to find frequent 1-itemset (itemset containing only 1 item)
- Use "set, reduce, map" at least once

In [5]:

```
# YOUR CODE MUST BE HERE
from functools import reduce
from collections import Counter
def gen_frequent_1_itemset(dataset, p):
    ls = list(reduce(lambda x, y: x + y, dataset)) # reduce
    count = list(Counter(ls).items())
    return map( lambda x :(x[0]) ,filter(lambda x: (x[1] / len(dataset)) >= p
```

In [6]:

```
# DO NOT EDIT THIS CELL
dataset = list(gen_record(data_str))
print(sorted(list(gen_frequent_1_itemset(dataset, 0.5))))
```

```
['apple', 'beer', 'milk', 'rice']
```

Your output must be (sorted list):

```
['apple', 'beer', 'milk', 'rice']
```

In [7]:

```
# DO NOT EDIT THIS CELL
dataset = list(gen_record(data_str))
print(sorted(list(gen_frequent_1_itemset(dataset, 0.7))))
```

```
['beer']
```

Your output must be (sorted list):

```
['beer']
```

In [8]:

```
# DO NOT EDIT THIS CELL
dataset = list(gen_record(data_str))
```

```
print(sorted(list(gen_frequent_1_itemset(dataset, 0.2))))
```

```
['apple', 'beer', 'chicken', 'mango', 'milk', 'rice']
```

Your output must be (sorted list):

```
['apple', 'beer', 'chicken', 'mango', 'milk', 'rice']
```

Problem 3 (10 pts)

- Define a function ***gen_frequent_2_itemset*** generating 2-itemset.
- It must be a generator.
- We want to find frequent 2-itemset (itemset containing only 2 items)
- Use "set, reduce, map" at least once

In [9]:

```
# YOUR CODE MUST BE HERE
from functools import reduce

def gen_frequent_2_itemset(dataset, p):
    t = list(map(lambda basket: tuple(map(lambda i: list(map(lambda j: (baske
    ls = (reduce(lambda x, y: x + y, t))
    lm = (reduce(lambda x, y: x + y, ls))
    count = list(Counter(lm).items())
    return map( lambda x :(x[0]) ,filter(lambda x: (x[1] / len(dataset)) >= p
```

In [10]:

```
# DO NOT EDIT THIS CELL
dataset = list(gen_record(data_str))
print(sorted(list(gen_frequent_2_itemset(dataset, 0.5))))
```

```
[('beer', 'rice')]
```

Your output must be:

```
[('beer', 'rice')]
```

In [11]:

```
# DO NOT EDIT THIS CELL
dataset = list(gen_record(data_str))
print(sorted(list(gen_frequent_2_itemset(dataset, 0.3))))
```

```
[('apple', 'beer'), ('beer', 'milk'), ('beer', 'rice')]
```

Your output must be:

```
[('apple', 'beer'), ('beer', 'milk'), ('beer', 'rice')]
```

In [12]:

```
# DO NOT EDIT THIS CELL
dataset = list(gen_record(data_str))
print(sorted(list(gen_frequent_2_itemset(dataset, 0.2))))
```

```
[('apple', 'beer'), ('apple', 'rice'), ('beer', 'chicken'), ('beer', 'milk'),
('beer', 'rice'), ('chicken', 'rice'), ('milk', 'rice')]
```

Your output must be:

```
[('apple', 'beer'), ('apple', 'rice'), ('beer', 'chicken'),
('beer', 'milk'), ('beer', 'rice'), ('chicken', 'rice'),
('milk', 'rice')]
```

Ethics:

If you cheat, you will get negative of the total points. If the homework total is 22 and you cheat, you get -22.

What to submit

- Run **all cells** after restarting the kernel
- Goto "File -> Print Preview"
- Print the page as pdf
- Pdf file name must be in a form of: homework_1_홍길동_202000001.pdf
- Submit the pdf file in google classroom
- No late homeworks will be accepted
- Your homework will be graded on the basis of correctness and programming skills