

Q3

Import Libraries

```
from pprint import pprint
import itertools
import math
```

(a) Explain what is frequent itemset and give an example of 2-itemset that is frequent itemset with support count = 8.

Frequent Itemset: Itemset is collection of one or more items. Frequent itemset is whose support is greater than or equal to a minimum support threshold defined.

```
# Transactions
transactions = [(1, frozenset({'B', 'D', 'F', 'H'})),
                (2, frozenset({'C', 'D', 'F', 'G'})),
                (3, frozenset({'A', 'D', 'F', 'G'})),
                (4, frozenset({'A', 'B', 'C', 'D', 'H'})),
                (5, frozenset({'A', 'C', 'F', 'G'})),
                (6, frozenset({'D', 'H'})),
                (7, frozenset({'A', 'B', 'E', 'F'})),
                (8, frozenset({'A', 'D', 'F', 'G', 'H'})),
                (9, frozenset({'A', 'C', 'D', 'F', 'G'})),
                (10, frozenset({'D', 'F', 'G', 'H'})),
                (11, frozenset({'A', 'C', 'D', 'E'})),
                (12, frozenset({'B', 'E', 'F', 'H'})),
                (13, frozenset({'D', 'F', 'G'})),
                (14, frozenset({'C', 'F', 'G', 'H'})),
                (15, frozenset({'A', 'C', 'D', 'F', 'H'})),
                ]
```

```
# unique items
items = set()
for i, x in transactions:
    items.update(x)
print('Total items in the data set - {}'.format(len(items)))
print('The items are - {}'.format(items))
```

Total items in the data set - 8

The items are - {'B', 'C', 'A', 'G', 'H', 'D', 'F', 'E'}

```
# Itemsets with support >= 1
itemsets = {}

for x in itertools.combinations(items, 2):
    for _, y in transactions:
        if set(x).issubset(y):
            if frozenset(x) not in itemsets:
                itemsets[frozenset(x)] = 0
            itemsets[frozenset(x)] += 1

itemsets_support8 = dict((itemset, count) for itemset, count in itemsets.items()
if count >= 8)
s = set(itemsets_support8.keys())
print('Itemsets with support = 8 are as following: ')
for x in s:
    print('[{}]'.format(', '.join(x)))
print('Total itemsets with support = 8 are ->
{}'.format(len(itemsets_support8)))
```

Itemsets with support = 8 are as following:

[F, G]

[D, F]

Total itemsets with support = 8 are -> 2

(b) Explain what is closed frequent itemset and list all of them with supportcount = 8.

Closed Frequent Itemset: It is a frequent itemset that is both closed and its support is greater than or equal to minsup. An itemset is closed in a data set if there exists no superset that has the same support count as this original itemset. Ref: http://www.hypertextbookshop.com/dataminingbook/public_version/contents/chapters/chapter002/section004/blue/page002.html

```
itemsets = {}
for i in range(1, len(items)+1):
    for x in itertools.combinations(items, i):
        for _, y in transactions:
            if set(x).issubset(y):
                if frozenset(x) not in itemsets:
```

```

        itemsets[frozenset(x)] = 0
        itemsets[frozenset(x)] += 1

itemsets

itemsets_support8_dict = dict((itemset, count) for itemset, count in
itemsets.items() if count >= 8)
itemsets_support8 = list(itemsets_support8_dict.keys())

# Find closed itemset
closed_frequent_itemsets = []
for i in range(len(itemsets_support8)):
    subsetFound = False
    for j in range(i+1, len(itemsets_support8)):

        if itemsets_support8[i].issubset(itemsets_support8[j]) and
itemsets_support8_dict[itemsets_support8[i]] <=
itemsets_support8_dict[itemsets_support8[j]]:
            subsetFound = True
            break

    if not subsetFound:
        closed_frequent_itemsets.append(itemsets_support8[i])

# print('itemsets_support8_dict: {}'.format(itemsets_support8_dict))
# s = set(itemsets_support8_dict.keys())
# for x in s:
#     print('[{}]'.format(', '.join(x)))

print('Closed Itemsets with support >= 8 are as following: ')
for x in closed_frequent_itemsets:
    print('[{}]'.format(', '.join(x)))
# print('Closed itemsets with support >= 8 are ->
{}'.format(len(itemsets_support8)))

```

```

Closed Itemsets with support >= 8 are as following:
[A]
[H]
[D]
[F]
[F, G]
[D, F]

```

(c) Explain what is maximal frequent itemset and list all of maximal itemset with support count = 8.

Maximal Frequent Itemset: It is a frequent itemset for which none of its immediate supersets are frequent. Ref: http://www.hypertextbookshop.com/dataminingbook/public_version/contents/chapters/chapter002/section004/blue/page001.html

```
itemsets = {}
for i in range(1, len(items)+1):
    for x in itertools.combinations(items, i):
        for _, y in transactions:
            if set(x).issubset(y):
                if frozenset(x) not in itemsets:
                    itemsets[frozenset(x)] = 0
                itemsets[frozenset(x)] += 1

itemsets

itemsets_support8_dict = dict((itemset, count) for itemset, count in
itemsets.items() if count >= 8)
itemsets_support8 = list(itemsets_support8_dict.keys())

# Find closed itemset
maximal_frequent_itemsets = []
for i in range(len(itemsets_support8)):
    subsetFound = False
    for j in range(i+1, len(itemsets_support8)):
        if itemsets_support8[i].issubset(itemsets_support8[j]) and
len(itemsets_support8[i])+1 == len(itemsets_support8[j]):
            subsetFound = True
            break

    if not subsetFound:
        maximal_frequent_itemsets.append(itemsets_support8[i])

# s = set(itemsets_support8_dict.keys())
# for x in s:
#     print('[{}]' .format(', '.join(x)))

print('Maximal Frequent Itemsets with support >= 8 are as following: ')
for x in maximal_frequent_itemsets:
    print('[{}]' .format(', '.join(x)))
# print('Closed itemsets with support >= 8 are ->
{}'.format(len(itemsets_support8)))
```

Maximal Frequent Itemsets with support ≥ 8 are as following:

```
[A]
[H]
[F, G]
[D, F]
```

(d) Compute the support and confidence for association rule $\{D, F\} \rightarrow \{G\}$

```
# Support: Fraction of transactions that contains {D,F,G}
association_rule_set = frozenset({'D', 'F', 'G'})
support = 0
# for i in range(len(association_rule_set), len(items)+1):
#     for x in itertools.combinations(items, i):
for _,t in transactions:
    # print('Transaction: {}'.format(t))
    if association_rule_set.issubset(t):
        support += 1
print('Support for association rule {{D, F}} -> {{G}} is
{}'.format(support/len(transactions)))

# Confidence
x_set = frozenset({'D', 'F'})
support_x = 0
for _,t in transactions:
    # print('Transaction: {}'.format(t))
    if x_set.issubset(t):
        support_x += 1
print('Confidence for association rule {{D, F}} -> {{G}} is
{}'.format(support/support_x))
```

```
Support for association rule {D, F} -> {G} is 0.4
Confidence for association rule {D, F} -> {G} is 0.75
```