Part 1

Exercise 1 - The Apriori Algorithm:

TID	*items_bought*
T100	{M, O, N, K, E, Y}
T200	{D, O, N, K, E, Y }
T300	{M, A, K, E}
T400	{M, U, C, K, Y}
T500	{C, O, O, K, I, E}

a)

1-itemsets:

{M}: 0.6
{O}: 0.6
{N}: Does not meet min_sup
{K}: 1.0
{E}: 0.8
{Y}: 0.6
{D}: Does not meet min_sup
{A}: Does not meet min_sup
{U}: Does not meet min_sup
{C}: Does not meet min_sup
{I}: Does not meet min_sup

2-itemsets:

Any 2-itemsets containing N, D, A, U, C, and I are eliminated by Apriori Property.

{M, O}: Does not meet min_sup
{M, K}: 0.6
{M, E}: Does not meet min_sup
{M, Y}: Does not meet min_sup
{O, K}: 0.6
{O, E}: 0.6
{O, Y}: Does not meet min_sup
{K, E}: 0.8
{K, Y}: 0.6
{E, Y}: Does not meet min_sup

3-itemsets:

Any 3-itemsets containing {M,O}, {M, E}, {M, Y}, {O, Y}, {E, Y}, {M, K} are eliminated by Apriori Property.

{O, K, E}: 0.6

b)

A closed frequent itemset is a set of items that appears frequently in a dataset and is not a subset of any other frequent itemset with the same frequency count. In other words, a closed frequent itemset is a set of items that has the maximum support among all the itemsets with the same set of items.

From the above list, {O, K, E}, {K, Y}, {M, K} are all closed frequent itemsets.

c)

A max frequent itemset is a set of items that appears frequently in a dataset and is not a subset of any other frequent itemset with a higher support count. In other words, a max frequent itemset is a set of items that has the maximum support count among all the itemsets that have the same items but with different cardinalities.

From the above list, {K} and {E} are the max frequent itemsets

What is absolute support?
Use number of items
Like 3!!!

The maximum frequent itemset is the itemset with the highest support in a dataset, where support refers to the proportion of transactions in the dataset that contain that itemset. In other words, it is the itemset that appears most frequently in the transactions in the dataset.
For example, consider a dataset of grocery store transactions, where each transaction contains a set of items purchased by a customer. If the itemset {bread, milk} appears in 40% of all transactions in the dataset, and no other itemset appears in a higher percentage of transactions, then {bread, milk} is the maximum frequent itemset in the dataset.
Finding the maximum frequent itemset is an important task in data mining and machine learning, as it can provide insights into the most popular combinations of items in a dataset, which can be useful for various applications such as product recommendation, market basket analysis, and customer segmentation.

d)

How to generate association rules:

1. Start with each frequent itemset of size 2 or more.
2. For each frequent itemset, generate all possible non-empty subsets of the itemset.
3. For each subset, compute the confidence and lift measures of the association rule that has the subset as the antecedent and the complement of the subset as the consequent.

4. If the confidence and lift measures exceed certain threshold values (e.g., 0.7 for confidence and 1.2 for lift), then the association rule is considered strong.

{M, K}:
    {K → M}:
        Confidence = 0.6 / 0.8 = 0.75
        Lift = 0.6 / (0.6 * 1.0) = 1.0
    {M → K}:
        Confidence = 0.6 / 0.6 = 1.0
        Lift = 0.6 / (0.6 * 1.0) = 1.0

{O, K}:
    O: 0.6, K: 1.0
    {O → K}:
        Confidence = 0.6 / 0.6 = 1.0
        Lift: 0.6 / (0.6 * 1.0) = 1.0
    {K → O}:
        Confidence = 0.6 / 1.0 = 0.6
        Lift: 0.6 / (1.0 * 0.6) = 1.0

{O, E}:
    O: 0.6, E: 0.8
    {O → E}:
        Confidence: 0.6 / 0.6 = 1.0
        Lift: 0.6 / (0.6 * 0.8) = 1.25
    {E → O}:
        Confidence: 0.6 / 0.8 = 0.75
        Lift: 0.6 / (0.8 * 0.6) = 1.25

{K, E}:
    K: 1.0, E: 0.8
    {K → E}:
        Confidence: 0.8 / 1.0 = 0.8
        Lift: 0.8 / (1.0 * 0.8) = 1
    {E → K}:
        Confidence: 0.8 / 0.8 = 1.0
        Lift: 0.8 / (0.8 * 1.0) = 1

{K, Y}:
    K: 1.0, Y: 0.6
    {K → Y}:
        Confidence: 0.6 / 1.0 = 0.6
        Lift: 0.6 / (1.0 * 0.6) = 1.0
    {Y → K}:
        Confidence: 0.6 / 0.6 = 1.0
        Lift = 0.6 / (0.6 * 1.0) = 1.0

{O, K, E}:
{O,K,E}: 0.6
{O,K}: 0.6, {K,E}: 0.8, {O,E}: 0.6
O: 0.6, K: 1.0, E: 0.8

{{O,K} → {E}}:
Confidence: 0.6 / 0.6 = 1.0
Lift: 0.6 / (0.6 * 0.8) = 1.25
{{E, K} → {O}}:
Confidence: 0.6 / 0.8 = 0.75
Lift: 0.6 / (0.8 * 0.6) = 1.25
{{O, E} → {K}}:
Confidence: 0.6 / 0.6 = 1.0
Lift: 0.6 / (0.6 * 1.0) = 1.0

```
confidence(A → B) = support(A ∪ B) / support(A)
lift(A → B) = support(A ∪ B) / (support(A) × support(B))

confidence({bread, milk} → {cheese}) = support({bread, milk, cheese}) /
support({bread, milk}) = 2 / 3 = 0.67

lift({bread, milk} → {cheese}) = support({bread, milk, cheese}) /
(support({bread, milk}) × support({cheese})) = 2 / (3 × 3/4) = 1.33
```

e)

{{O,K} → {E}}:
Confidence: 0.6 / 0.6 = 1.0
Lift: 0.6 / (0.6 * 0.8) = 1.25
{O → E}:
Confidence: 0.6 / 0.6 = 1.0
Lift: 0.6 / (0.6 * 0.8) = 1.25

This is the strongest rule output as it has the highest confidence and lift and contains the most number of items for all frequent itemsets that have the same measures. Additionally, we can say that the rule {O → E} is also a strong rule because it has the same level of confidence and lift as {{O,K} → {E}.

Exercise 2 - The FP-Growth Algorithm:

a) F-List

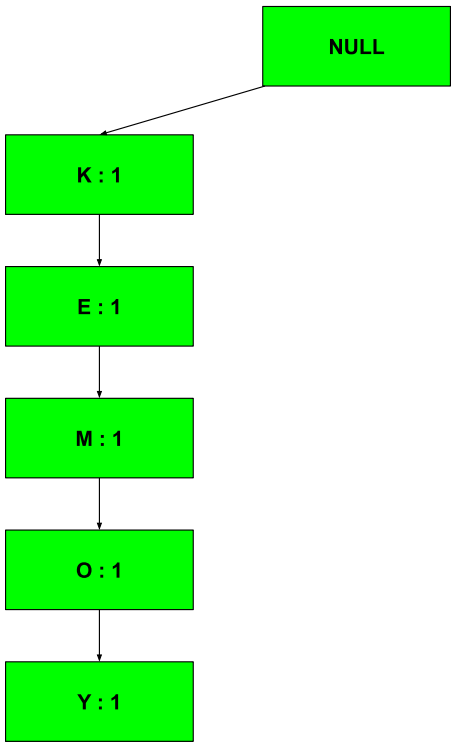
{K}: 1.0, 5
{E}: 0.8, 4
{M, O, Y}: 0.6, 3

{C, N}: 0.4, 2
{A, D, I, U}: 0.2, 1

Frequent Pattern set = {K : 5, E : 4, M : 3, O : 3, Y : 3}

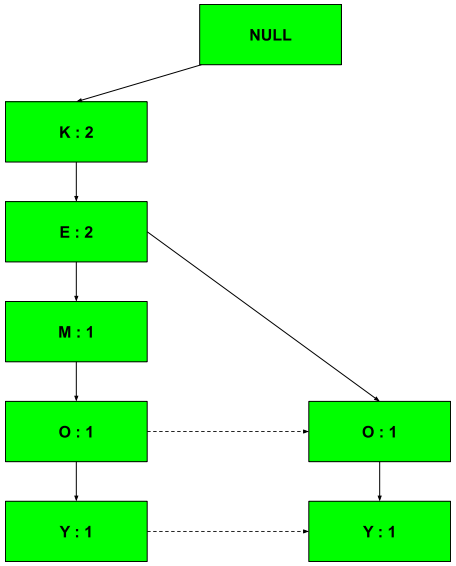
Transaction ID	Items	Ordered-Item Set
100	{M, O, N, K, E, Y}	{K, E, M, O, Y}
200	{D, O, N, K, E, Y}	{K, E, O, Y}
300	{M, A, K, E}	{K, E, M}
400	{M, U, C, K, Y}	{K, M, Y}
500	{C, O, O, K, I, E}	{K, E, O}

b) Initial FP-Tree

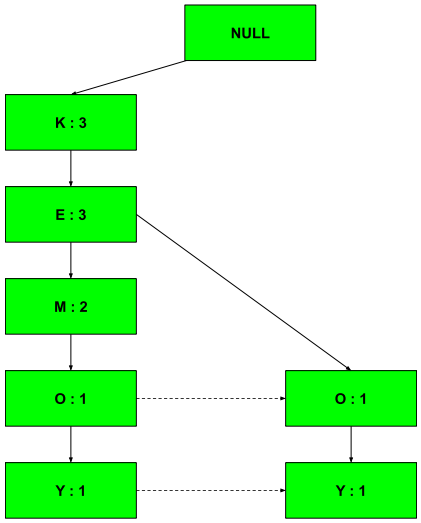


c) Executing FP_growth()

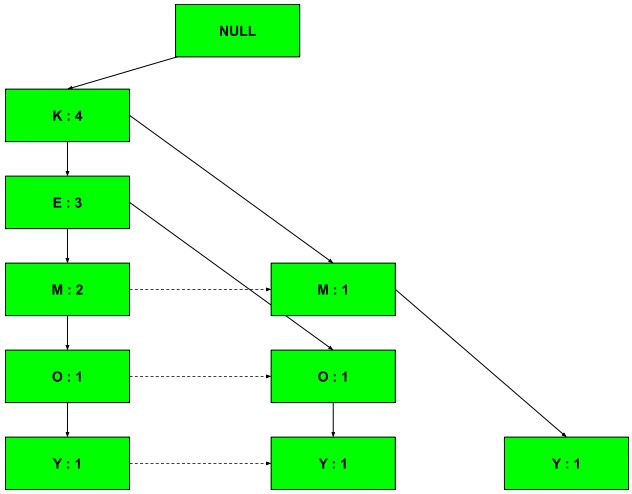
$tree_{\{K, E, O, Y\}}$



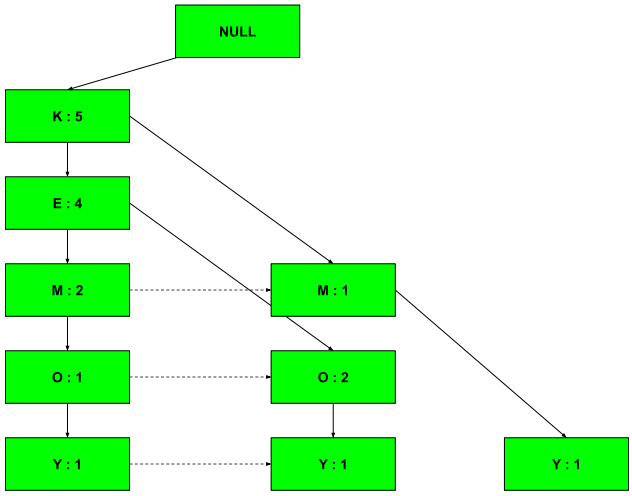
$tree_{\{K, E, M\}}$



$tree_{\{K, M, Y\}}$



$tree_{\{K, E, O\}}$



Item	Conditional Pattern Base	Conditional Frequent Pattern Tree	Frequent Pattern Generated
Y	{K, E, M, O : 1}, {K, E, O : 1}, {K, M : 1}	{K : 3}	{ K, Y : 3 }
O	{K, E, M : 1}, {K, E : 2}	{K, E : 3}	{ (K, O : 3), (E, O : 3), (E, K, O : 3) }
M	{K, E : 2}, {K : 1}	{K : 3}	{ K, M : 3 }
E	{K : 4}	{K : 4}	{ E , K : 3 }
K			

d) Comparing Apirori and FP-growth

It definitely takes less space to use the FP-growth algorithm to find frequent patterns, but it certainly took us a lot more time to compute manually compared to the Apriori algorithm.

Exercise 3 - The Eclat Algorithm:

a) Vertical Data Format

Transaction ID	Item	Frequency
100	M	1
100	O	1
100	N	1
100	K	1
100	E	1
100	Y	1
200	D	1
200	O	1
200	N	1
200	K	1
200	E	1
200	Y	1
300	M	1
300	A	1
300	K	1
300	E	1
400	M	1
400	U	1
400	C	1
400	K	1
400	Y	1
500	C	1
500	O	2
500	K	1

500	I	1
500	E	1

b) ECLAT algorithm with minimum support threshold set to 2:

Item	Support Count
K	5
E	4
O	4
M	3
Y	3
C	2
N	2
A	1
D	1
I	1
U	1
K, E	4
K, M	3
K, O	3
K, Y	3
O, E	3
C, K	2
K, N	2
N, Y	2
N, E	2
O, N	2
O, Y	2

M, E	2
M, Y	2
E, Y	2

Exercise 4:

	A	**NOT A**
B	65	40
NOT B	35	10

a) Compute the support and confidence for {A → B}

```
confidence(A → B) = support(A ∪ B) / support(A)
```

{A → B}: Support val for: A = 0.35, B = 0.40
Support: 0.65
Confidence: 0.65 / 0.35 = 1.86

Answer: Yes, this is a moderately strong rule.

b) Compute the lift for {A → B}

```
lift(A → B) = support(A ∪ B) / (support(A) × support(B))
```

{A → B}:
Lift: 0.65 / (0.35 * 0.40) = 4.64

Answer: This lift level shows how much the occurrence of A is dependent on B.

c) Compute the expected values:

To compute the expected values for each observed value in the contingency table, we can use the following formula:
Eij = (Ai * Bj) / N
where Eij is the expected count for cell (i,j), Ai is the total count for row i, Bj is the total count for column j, and N is the total count of all observations.

For (1,1):
Observed: 65
Expected: E_{ij} = (100*105) / 150 = 70

For (1, 2):
Observed: 40
Expected: E_{ij} = (50 * 105) / 150 = 35

For (2, 1):
Observed: 35
Expected: E_{ij} = (100 * 45) / 150 = 30

For (2, 2):
Observed: 10
Expected: E_{ij} = (50 * 45) / 150 = 15

d)

$X2 = \Sigma((O-E)2 / E)$

Where Σ is the sum over all cells in the contingency table, O is the observed count in a cell, and E is the expected count in that cell.

$X2 = (((65-70)*2/70) + ((40-35)*2/35) + ((35-30)*2/30) + ((10-15)*2/15)) = -0.19$

This does not imply dependency among A because of how low the value is

e) Rule {A → NOT B}, what is the confidence

f) Kulczynski(A, B) = $|A \cap B| / (|A| + |B| - |A \cap B|)$