1.

Generate table (candidate set) of support count for each item

a	6
b	8
c	6
d	4
e	2
I	1
J	1
k	1

Get rid of those less than min_support (2) for l1

a	6
b	8
c	6
d	4
e	2

Join step for l2:

A, b	4
A, c	4
A, d	2
A,e	2
B, c	4
B,d	3
B,e	2
C,d	1
C, e	1
D, e	0

Remove those with less than 2:

A, b	4
A, c	4
A, d	2
A,e	2
B, c	4
B,d	3
B,e	2

Join step for l3:

A, b,c	2
A, b,d	2
A,b,e	2
A, c,d	1
A, C ,E	1
A, D, E	0
B C D	1
B C E	1
B D E	0

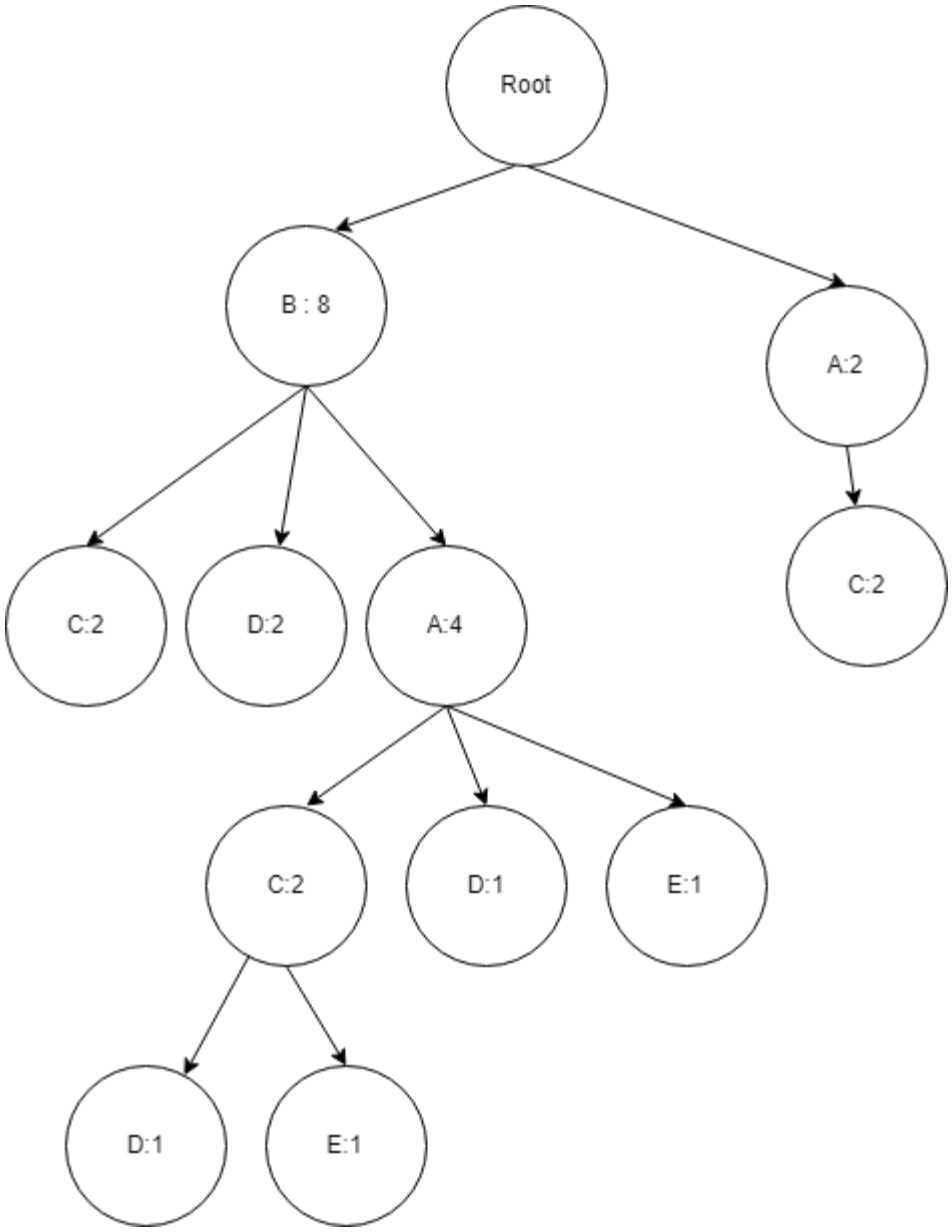
Remove those with less than 2

A, b,c	2
A, b,d	2
A,b,e	2

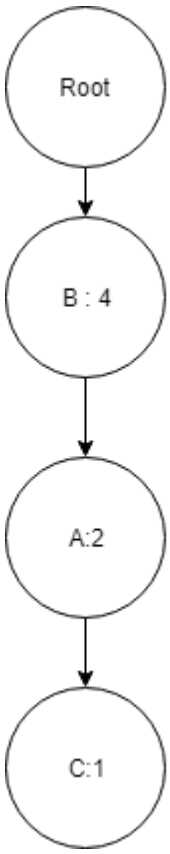
FP-tree: Frequency over min-support

b	8
a	6
c	6
d	4
e	2

Ordered List
B,c
B,a,d
A,c
B,d
B,a,c,e
B,c
A,c
b,a,e
B,d
B,a,c,d



Conditional data base : (B:2) (B:1 A:1 D:1) (B:1 A:1 C:1 D:1)



2. Output:

== toy.txt
min_support: 0.6 min_conf: 0.8
Running ...

---------- FREQUENT PATTERNS
Itemset: "xxx", "zzz" , 60.00%
Itemset: "yyy", "zzz" , 60.00%
Itemset: "xxx", "yyy", "zzz" , 60.00%
Itemset: "xxx", "yyy" , 80.00%
----> 4 printed

---------- RULES:
Rule: "xxx" ==> "yyy" , 0.800
Rule: "yyy" ==> "xxx" , 1.000
Rule: "zzz" ==> "xxx" , 1.000
Rule: "zzz" ==> "yyy" , 1.000
Rule: "zzz" ==> "xxx", "yyy" , 1.000
Rule: "xxx", "zzz" ==> "yyy" , 1.000
Rule: "yyy", "zzz" ==> "xxx" , 1.000
----> 7 printed

0.00699996948242 sec

== user_movies.txt
min_support: 0.26 min_conf: 0.68
Running ...

---------- FREQUENT PATTERNS
Itemset: "Forrest Gump (1994)", "Braveheart (1995)" , 27.33%
Itemset: "Forrest Gump (1994)", "Shawshank Redemption, The (1994)" , 28.35%
Itemset: "Pulp Fiction (1994)", "Braveheart (1995)" , 28.57%
Itemset: "Star Wars: Episode IV - A New Hope (1977)", "Star Wars: Episode V - The Empire Strikes Back (1980)" , 28.57%
Itemset: "Pulp Fiction (1994)", "Forrest Gump (1994)" , 30.10%
Itemset: "Pulp Fiction (1994)", "Shawshank Redemption, The (1994)" , 31.63%
----> 6 printed

---------- RULES:
Rule: "Braveheart (1995)" ==> "Forrest Gump (1994)" , 0.686
Rule: "Forrest Gump (1994)" ==> "Pulp Fiction (1994)" , 0.686
Rule: "Braveheart (1995)" ==> "Pulp Fiction (1994)" , 0.717
Rule: "Shawshank Redemption, The (1994)" ==> "Pulp Fiction (1994)" , 0.725
Rule: "Star Wars: Episode V - The Empire Strikes Back (1980)" ==> "Star Wars: Episode IV - A New Hope (1977)" , 0.897
----> 5 printed

1.53900003433 sec

== movie_tags.txt
min_support: 0.0028 min_conf: 0.6
Running ...

---------- FREQUENT PATTERNS
Itemset: "dark comedy", "black comedy" , 0.28%
Itemset: "funny", "quirky" , 0.28%
Itemset: "future", "sci-fi" , 0.28%
Itemset: "fantasy", "adventure" , 0.29%
Itemset: "comic book", "superhero" , 0.30%
Itemset: "atmospheric", "based on a book" , 0.31%
Itemset: "Nudity (Topless)", "nudity (topless)" , 0.33%
Itemset: "sci-fi", "space" , 0.34%
Itemset: "dystopia", "sci-fi" , 0.37%
Itemset: "imdb top 250", "atmospheric" , 0.37%
Itemset: "aliens", "sci-fi" , 0.38%
Itemset: "imdb top 250", "classic" , 0.39%
Itemset: "action", "sci-fi" , 0.39%
Itemset: "stylized", "atmospheric" , 0.47%
Itemset: "funny", "comedy" , 0.57%

Itemset: "based on a book", "adapted from:book" , 0.60%
----> 16 printed

---------- RULES:
Rule: "future" ==> "sci-fi" , 0.729
----> 1 printed

50.9389998913 sec

2.1. Because every global frequent pattern is mined from local frequent pattern, its guaranteed that it is at least a local frequent pattern
2.2. Because there are times where we delete local frequent patterns, such as if it does not have more than the minimum support, it is not guaranteed a local frequent pattern is a global frequent pattern.
3.
Confidence: Beers -> nuts = P(nuts | beer) = support (beer and nuts) / support(nuts)
Confidence: 150/850 = 0.1764
Lift: P(Beers or Nuts) / P(beer) / P(nuts) = (150/10000)/ (500/10000) / (850/10000) = 3.5294
Chi-square:
Expected values: 850*500/10000, 9500*850/10000, 500*9150/10000, 9150*9500/10000
42.5, 807.5, 475, 8692.5
(150-42.5)^2/10000 + (807.5-700)^2/10000 + (475-350)^2/10000 + (8800-8692.5)^2/10000
=5.029
All_confidence:
Min(p(nuts|beer) and p(beer|nuts)) = 150/500 = .3
From these statistics we can tell that beers and nuts are not closely correlated.
4. Contains 4 events/ elements, but has a length of 6.
Subsequences:
Length of 1: 6
Length of 2: 5+4 + 2+2 =13
Length of 3: 4+ 2+ 2 = 8
Length of 4: 4
Total: 31
b.
 <ace>, take out first element = <ce>, no other element with last element removed is <ce>
<bcd>, take out element and it = <cd>, no other element with last element removed is <cd>
<bce> => <ce> no other element
<acd> => <cd> no other element
<abd> => <bd> no other element
<(ab)c> => <bc>, elements <b(cd)> and <bce> can be joined
Thus we generate C4 which is = <(ab)(cd)> and <(ab)ce>
Pruning:
Generate subsequences from C4:
From <abcd>, => <abc>, <abd>, <acd> <bcd>
All of these are within L3 so <(ab) (cd)> is in C4.
From <abce> = > <abc>, <abe>, <ace>, <bce>
Since <abe> is not in L3, <abce> is not a valid candidate.

C4: <(ab)(cd)>