CSIT5210

FP-Tree

Prepared by Raymond Wong
Presented by Raymond Wong
raywong@cse

Large Itemset Mining

- Frequent Itemset Mining

Problem: to find all "large" (or frequent) itemsets
with support at least a threshold
(i.e., itemsets with support >= 3)

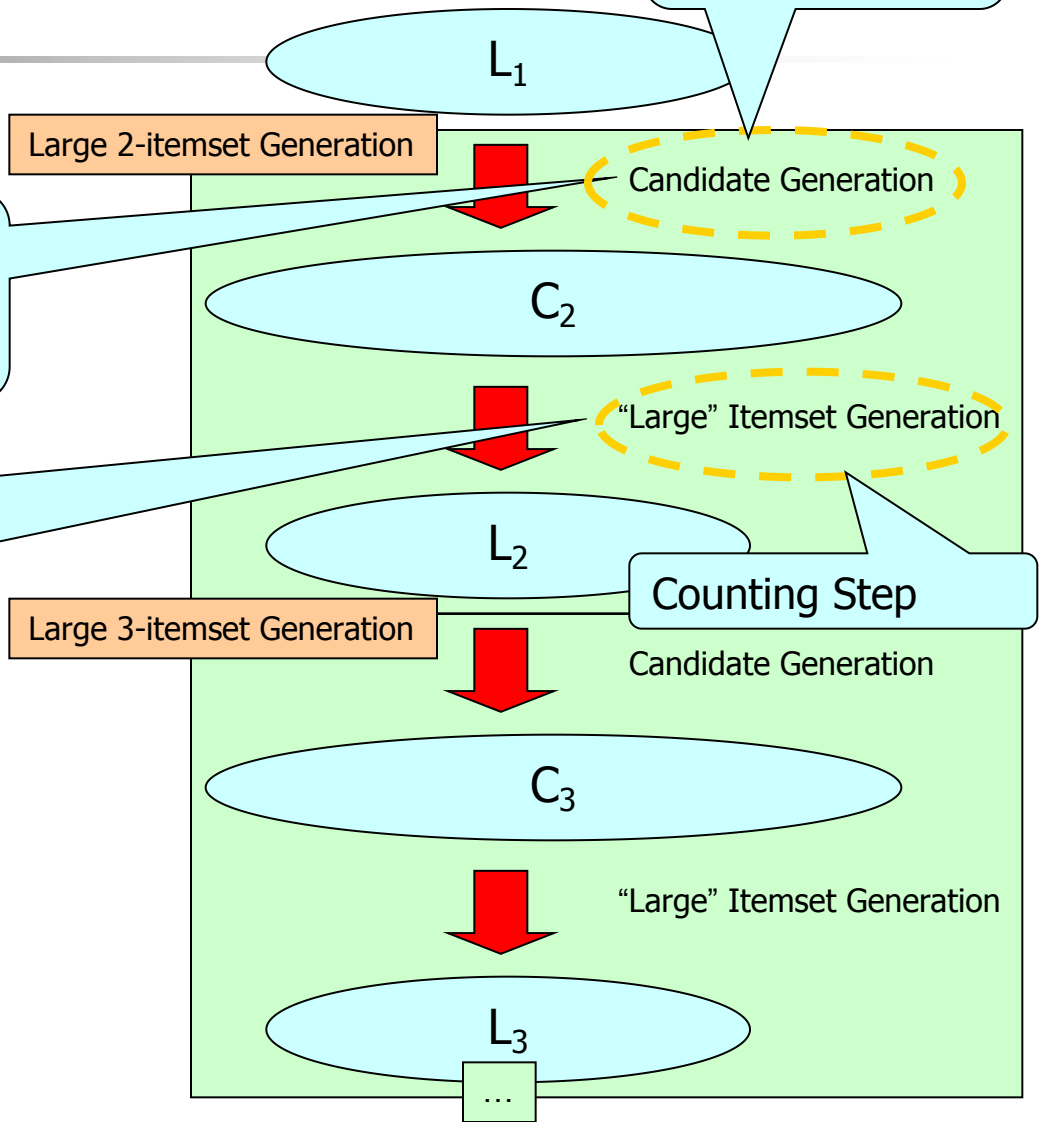
TID	Items Bought
100	a, b, c, d, e, f, g, h
200	a, f, g
300	b, d, e, f, j
400	a, b, d, i, k
500	a, b, e, g

Apriori

1. Join Step
2. Prune Step

Disadvantage 1: It is costly to handle a large number of candidate sets

Disadvantage 2: It is tedious to repeatedly scan the database and check the candidate patterns



FP-tree

- Scan the database once to store all essential information in a data structure called FP-tree (Frequent Pattern Tree)
- The FP-tree is concise and is used in directly generating large itemsets

FP-tree

Step 1: Deduce the ordered frequent items. For items with the same frequency, the order is given by the alphabetical order.

Step 2: Construct the FP-tree from the above data

Step 3: From the FP-tree above, construct the FP-conditional tree for each item (or itemset).

Step 4: Determine the frequent patterns.

FP-tree

- Frequent Itemset Mining

Problem: to find all "large" (or frequent) itemsets
with support at least a threshold
(i.e., itemsets with support >= 3)

TID	Items Bought
100	a, b, c, d, e, f, g, h
200	a, f, g
300	b, d, e, f, j
400	a, b, d, i, k
500	a, b, e, g

FP-tree

TID	Items Bought
100	a, b, c, d, e, f, g, h
200	a, f, g
300	b, d, e, f, j
400	a, b, d, i, k
500	a, b, e, g

TID	Items Bought
100	a, b, c, d, e, f, g, h
200	a, f, g
300	b, d, e, f, j
400	a, b, d, i, k
500	a, b, e, g

TID	Items Bought	(Ordered) Frequent Items
100	a, b, c, d, e, f, g, h	
200	a, f, g	
300	b, d, e, f, j	
400	a, b, d, i, k	
500	a, b, e, g	

Threshold = 3

Item	Frequency
a	4
b	
c	
d	
e	
f	
g	
h	
i	
j	
k	

TID	Items Bought	(Ordered) Frequent Items
100	a, b, c, d, e, f, g, h	
200	a, f, g	
300	b, d, e, f, j	
400	a, b, d, i, k	
500	a, b, e, g	

Threshold = 3

Item	Frequency
a	4
b	4
c	1
d	3
e	3
f	3
g	3
h	1
i	1
j	1
k	1

TID	Items Bought	(Ordered) Frequent Items
100	a, b, c, d, e, f, g, h	
200	a, f, g	
300	b, d, e, f, j	
400	a, b, d, i, k	
500	a, b, e, g	

Threshold = 3

Item	Frequency
a	4
b	4
c	1
d	3
e	3
f	3
g	3
h	1
i	1
j	1
k	1

Item	Frequency
a	4
b	4
d	3
e	3
f	3
g	3

TID	Items Bought	(Ordered) Frequent Items
100	a, b, c, d, e, f, g, h	a, b, d, e, f, g
200	a, f, g	a, f, g
300	b, d, e, f, j	b, d, e, f
400	a, b, d, i, k	a, b, d
500	a, b, e, g	a, b, e, g

Threshold = 3

Item	Frequency
a	4
b	4
c	1
d	3
e	3
f	3
g	3
h	1
i	1
j	1
k	1

Item	Frequency
a	4
b	4
d	3
e	3
f	3
g	3

FP-tree

Step 1: Deduce the ordered frequent items. For items with the same frequency, the order is given by the alphabetical order.

Step 2: Construct the FP-tree from the above data

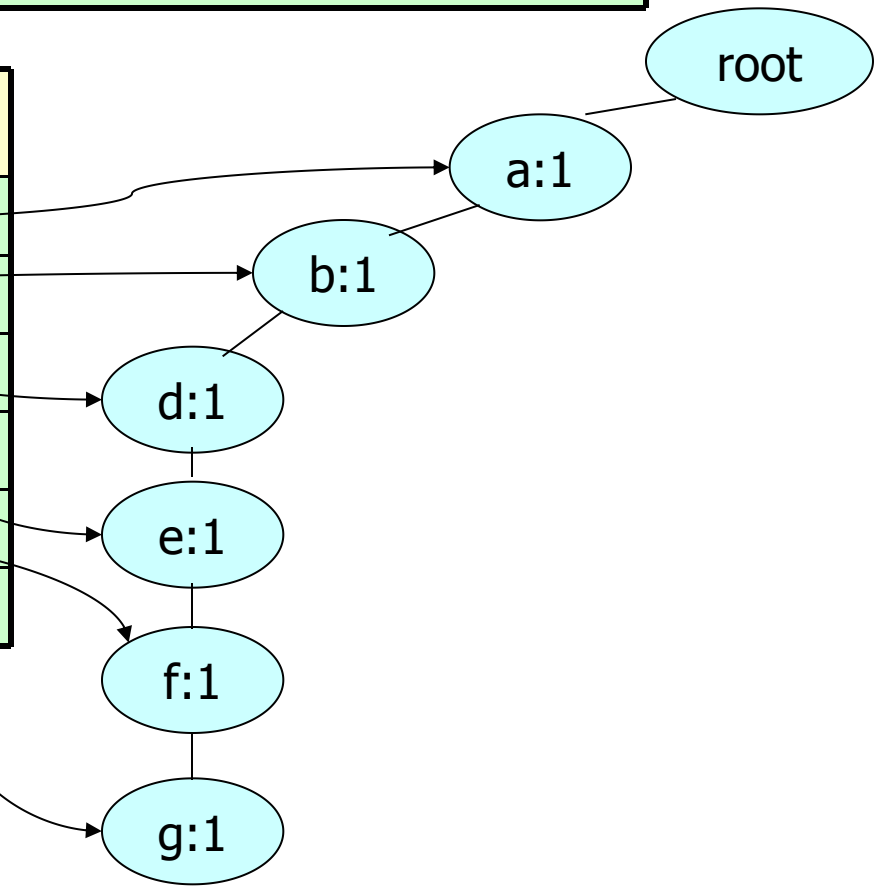
Step 3: From the FP-tree above, construct the FP-conditional tree for each item (or itemset).

Step 4: Determine the frequent patterns.

TID	Items Bought	(Ordered) Frequent Items
100	a, b, c, d, e, f, g, h	a, b, d, e, f, g
200	a, f, g	a, f, g
300	b, d, e, f, j	b, d, e, f
400	a, b, d, i, k	a, b, d
500	a, b, e, g	a, b, e, g

Threshold = 3

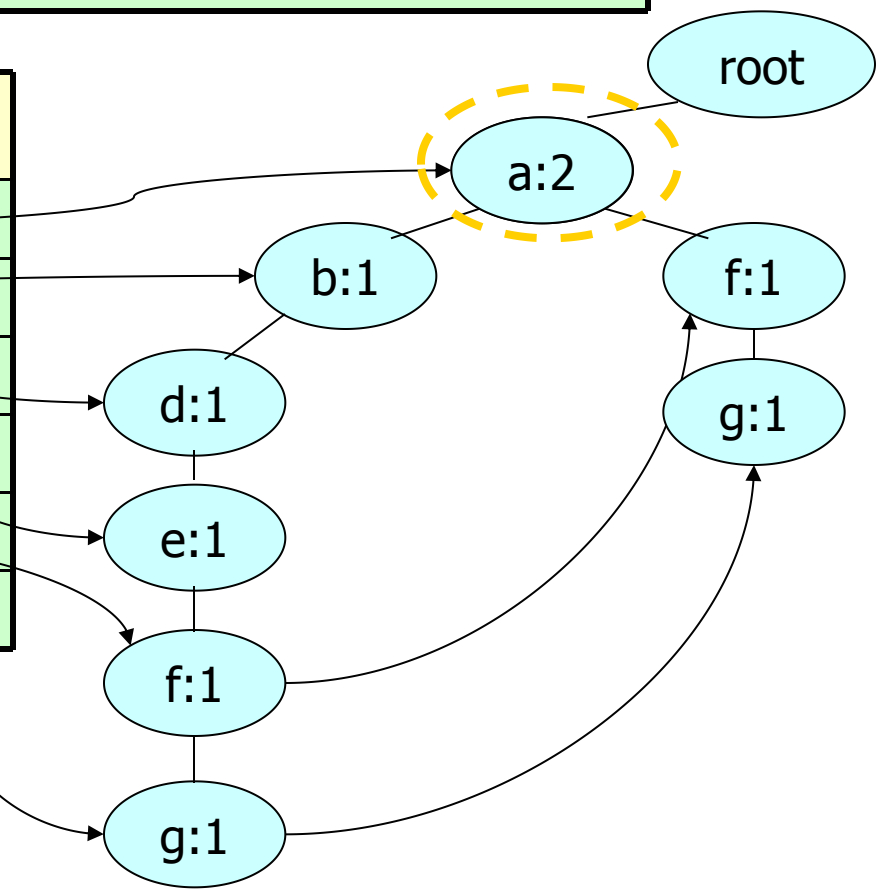
Item	Head of node-link
a	
b	
d	
e	
f	
g	



TID	Items Bought	(Ordered) Frequent Items
100	a, b, c, d, e, f, g, h	a, b, d, e, f, g
200	a, f, g	a, f, g
300	b, d, e, f, j	b, d, e, f
400	a, b, d, i, k	a, b, d
500	a, b, e, g	a, b, e, g

Threshold = 3

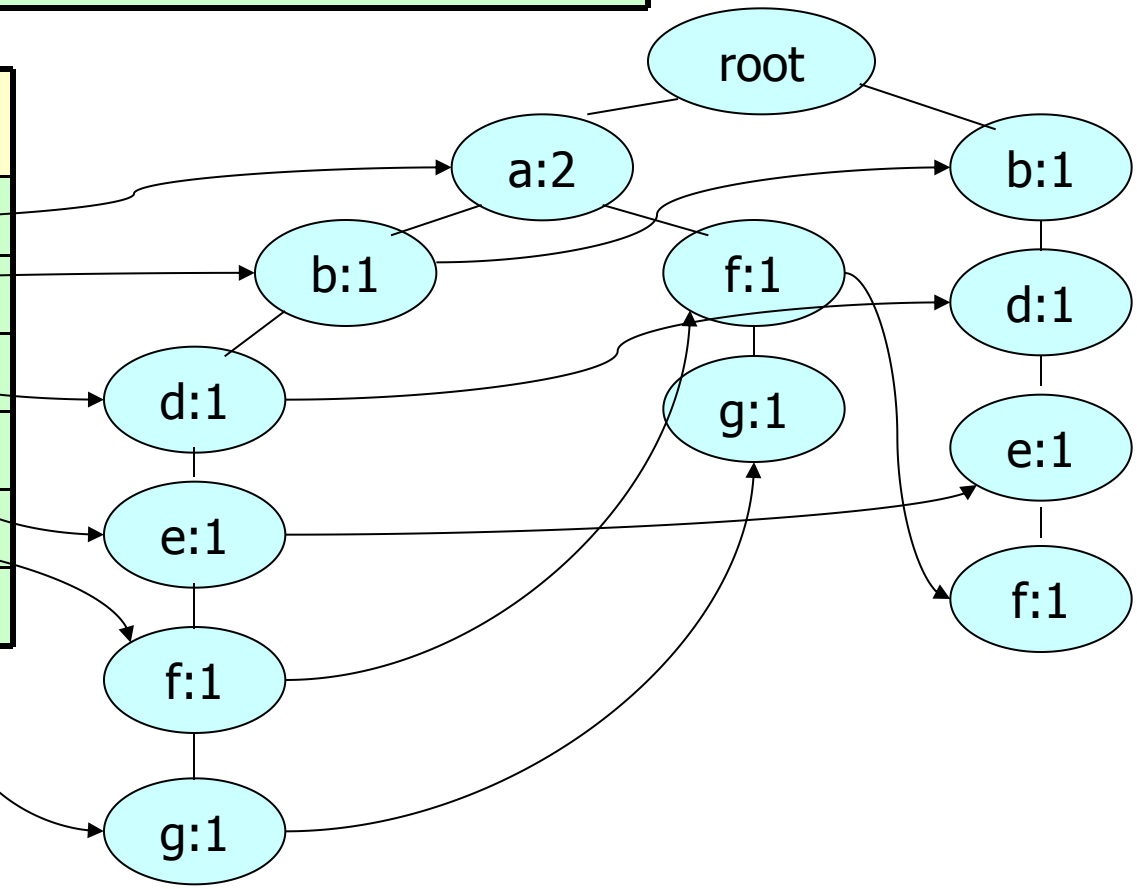
Item	Head of node-link
a	
b	
d	
e	
f	
g	



TID	Items Bought	(Ordered) Frequent Items
100	a, b, c, d, e, f, g, h	a, b, d, e, f, g
200	a, f, g	a, f, g
300	b, d, e, f, j	b, d, e, f
400	a, b, d, i, k	a, b, d
500	a, b, e, g	a, b, e, g

Threshold = 3

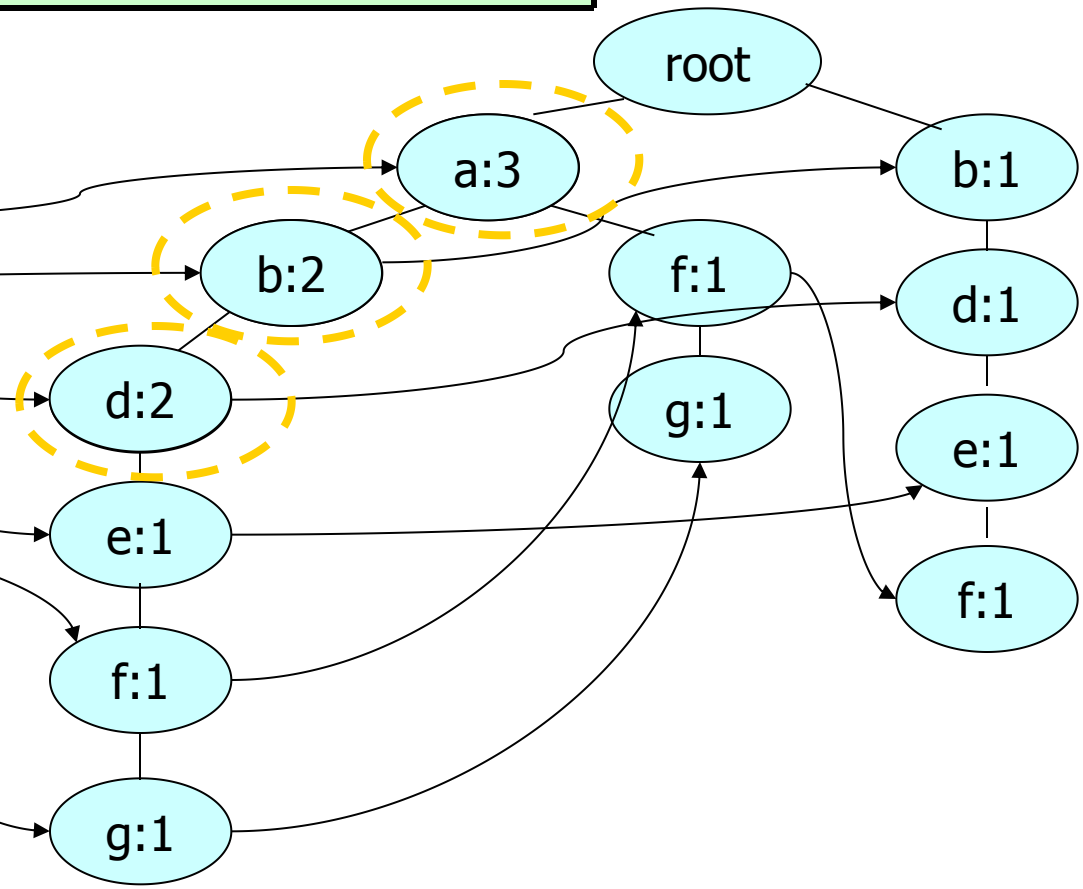
Item	Head of node-link
a	
b	
d	
e	
f	
g	



TID	Items Bought	(Ordered) Frequent Items
100	a, b, c, d, e, f, g, h	a, b, d, e, f, g
200	a, f, g	a, f, g
300	b, d, e, f, j	b, d, e, f
400	a, b, d, i, k	a, b, d
500	a, b, e, g	a, b, e, g

Threshold = 3

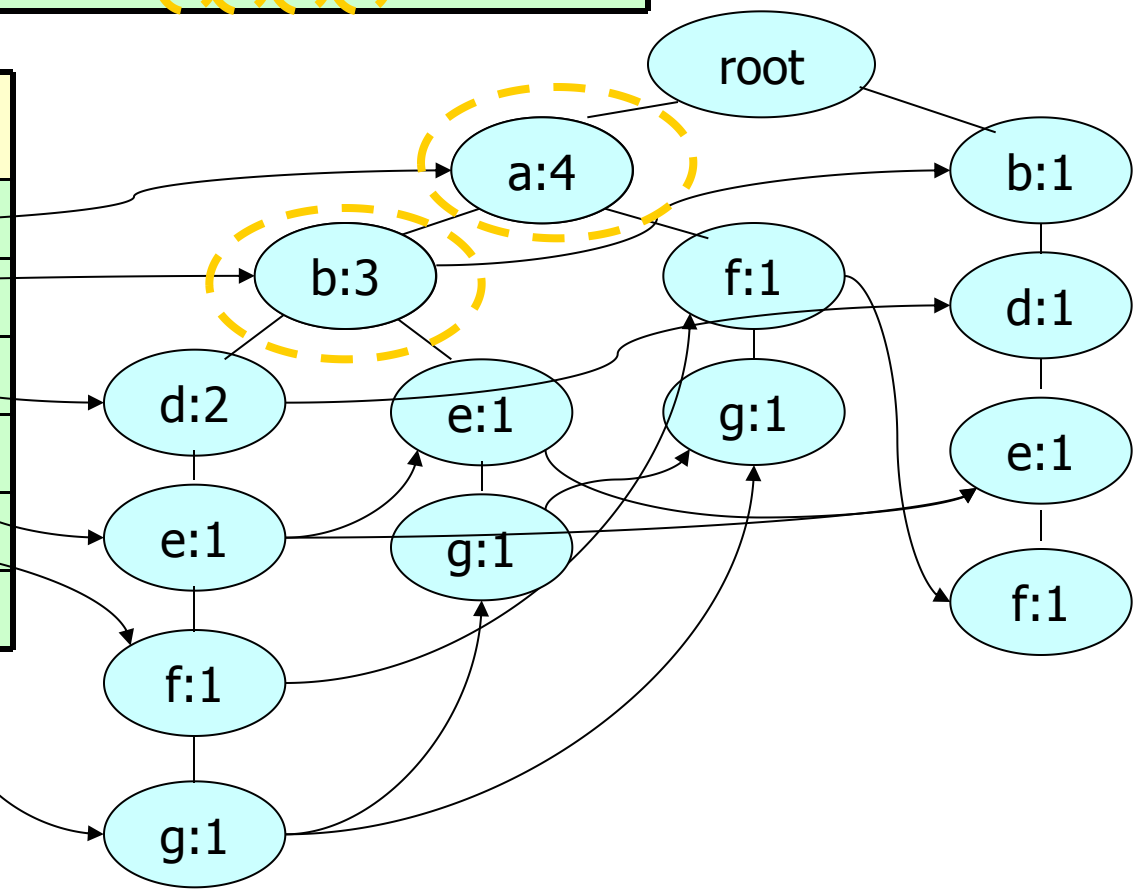
Item	Head of node-link
a	
b	
d	
e	
f	
g	



TID	Items Bought	(Ordered) Frequent Items
100	a, b, c, d, e, f, g, h	a, b, d, e, f, g
200	a, f, g	a, f, g
300	b, d, e, f, j	b, d, e, f
400	a, b, d, i, k	a, b, d
500	a, b, e, g	a, b, e, g

Threshold = 3

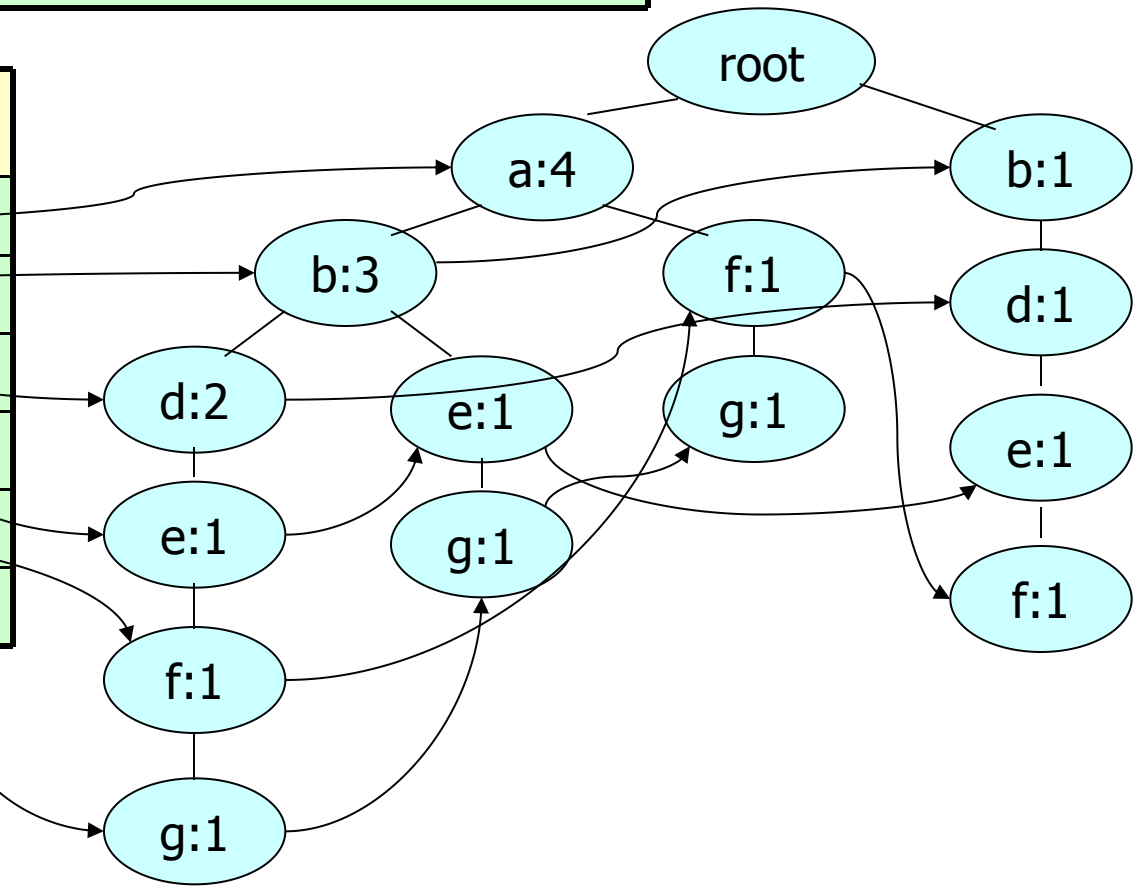
Item	Head of node-link
a	
b	
d	
e	
f	
g	



TID	Items Bought	(Ordered) Frequent Items
100	a, b, c, d, e, f, g, h	a, b, d, e, f, g
200	a, f, g	a, f, g
300	b, d, e, f, j	b, d, e, f
400	a, b, d, i, k	a, b, d
500	a, b, e, g	a, b, e, g

Threshold = 3

Item	Head of node-link
a	
b	
d	
e	
f	
g	



FP-tree

Step 1: Deduce the ordered frequent items. For items with the same frequency, the order is given by the alphabetical order.

Step 2: Construct the FP-tree from the above data

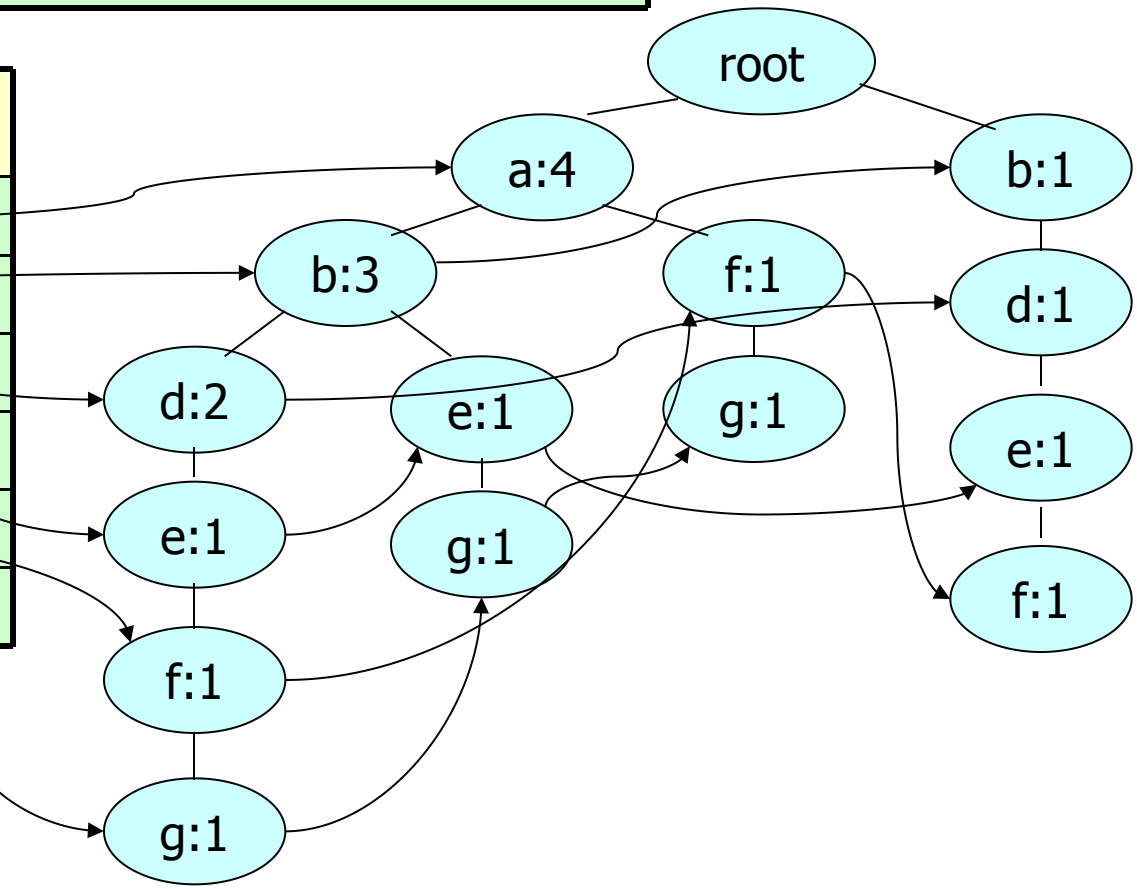
Step 3: From the FP-tree above, construct the FP-conditional tree for each item (or itemset).

Step 4: Determine the frequent patterns.

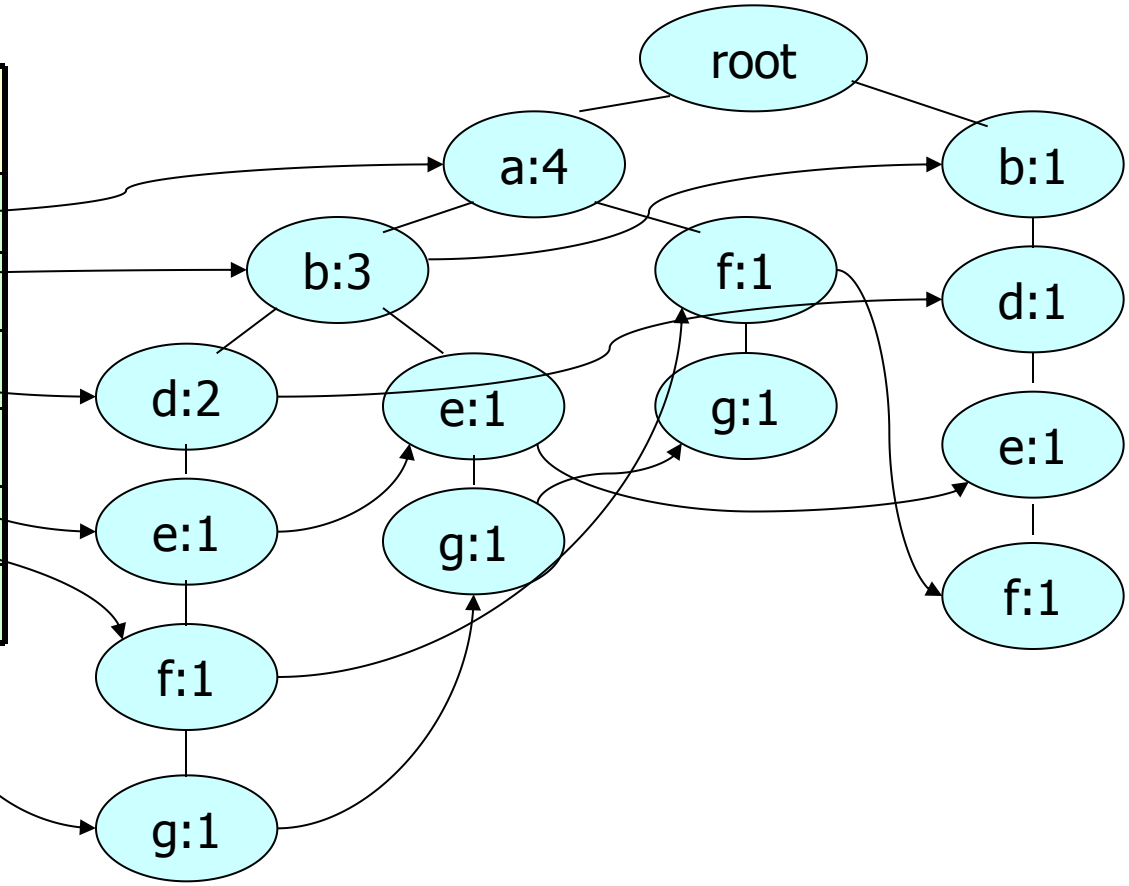
TID	Items Bought	(Ordered) Frequent Items
100	a, b, c, d, e, f, g, h	a, b, d, e, f, g
200	a, f, g	a, f, g
300	b, d, e, f, j	b, d, e, f
400	a, b, d, i, k	a, b, d
500	a, b, e, g	a, b, e, g

Threshold = 3

Item	Head of node-link
a	
b	
d	
e	
f	
g	

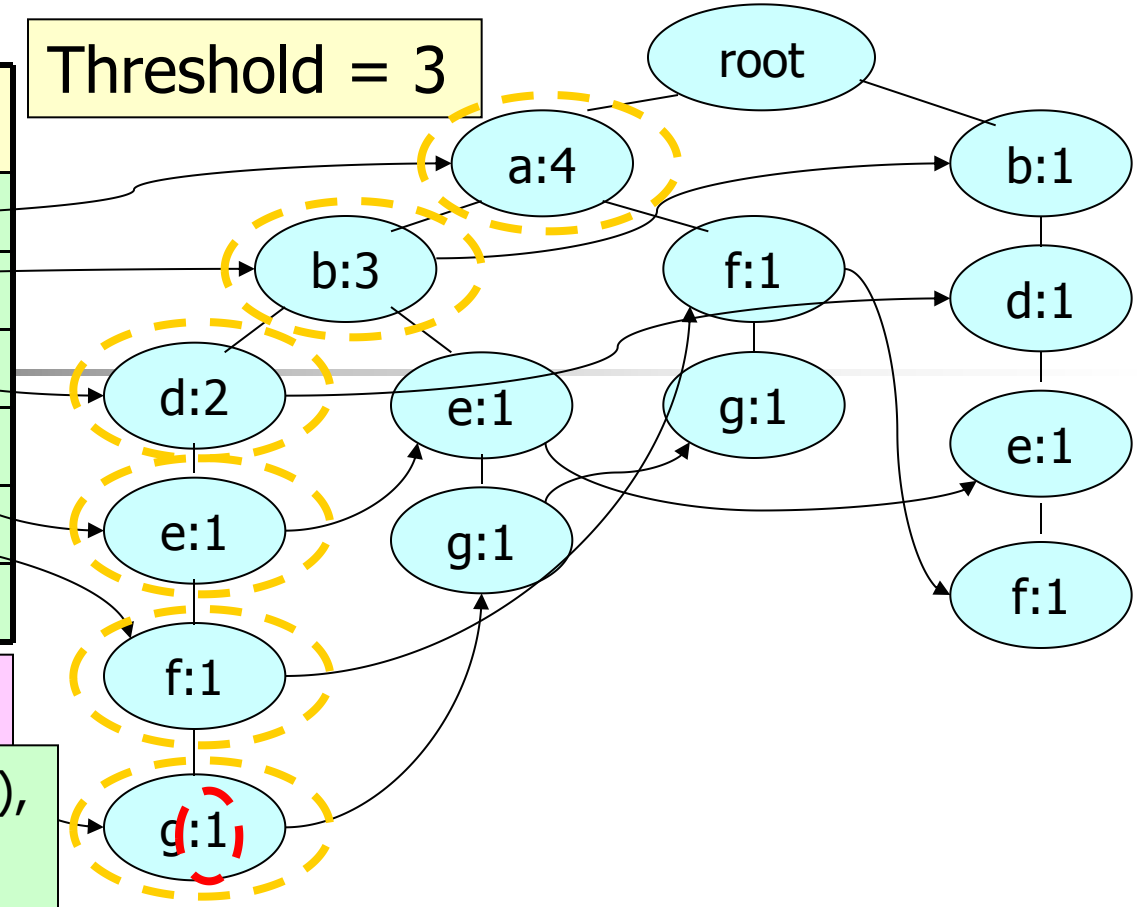


Item	Head of node-link
a	
b	
d	
e	
f	
g	



Threshold = 3

Item	Head of node-link
a	
b	
d	
e	
f	
g	



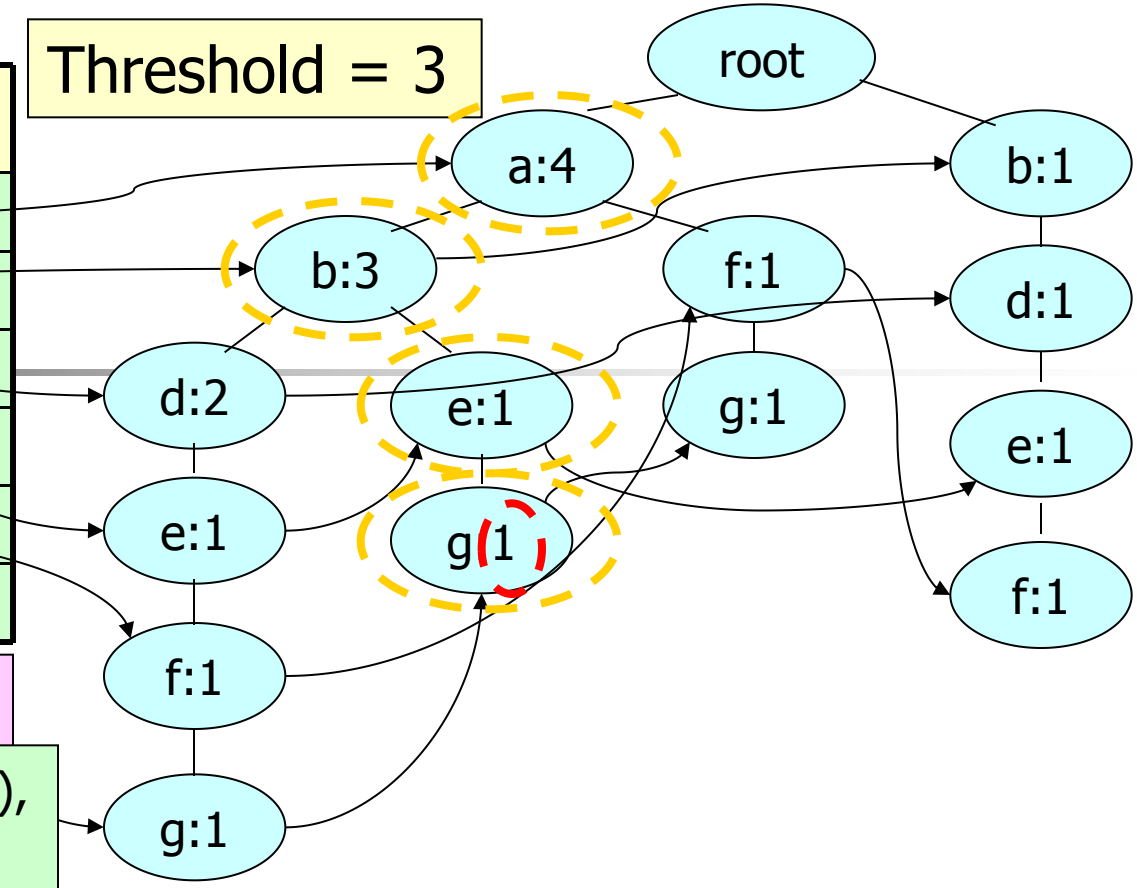
Cond. FP-tree on “g”

{ (a:1, b:1, d:1, e:1, f:1, g:1),

}

Item	Head of node-link
a	
b	
d	
e	
f	
g	

Threshold = 3

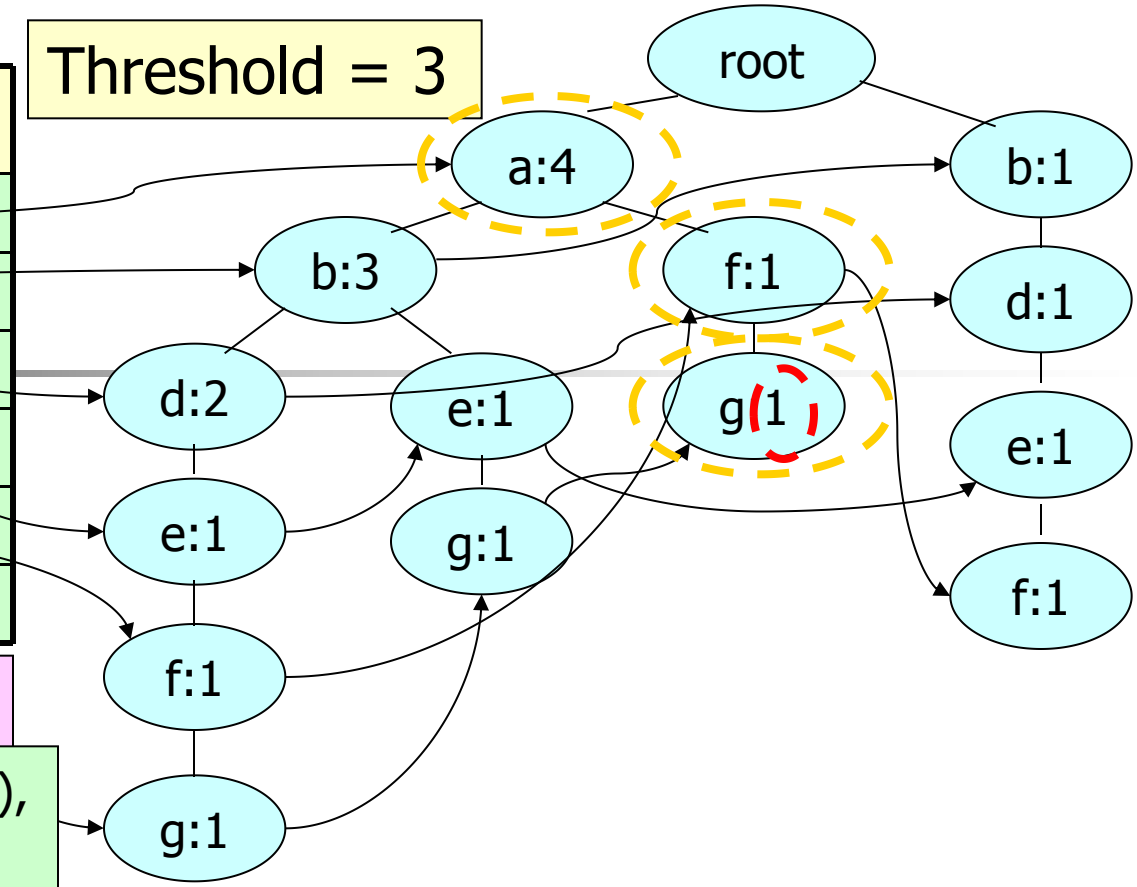


Cond. FP-tree on “g”

{ (a:1, b:1, d:1, e:1, f:1, g:1),
(a:1, b:1, e:1, g:1),
}

Threshold = 3

Item	Head of node-link
a	
b	
d	
e	
f	
g	



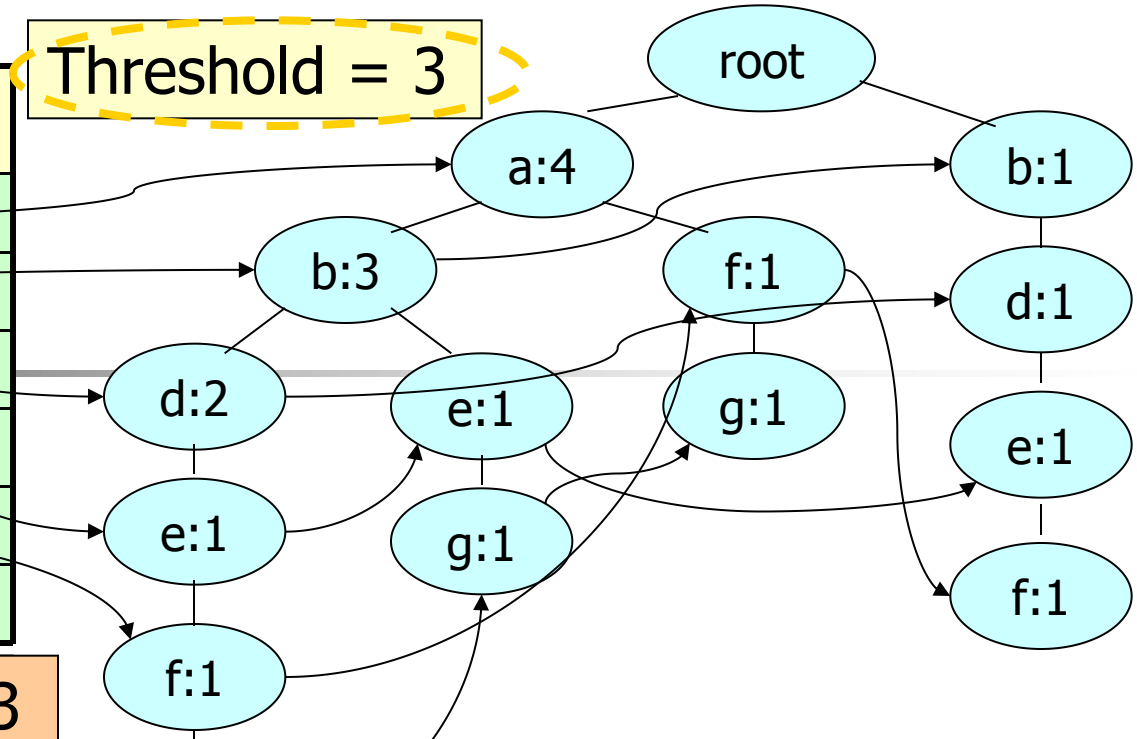
Cond. FP-tree on “g”

{ (a:1, b:1, d:1, e:1, f:1, g:1),
(a:1, b:1, e:1, g:1),
(a:1, f:1, g:1)}

Item	Frequency
a	3
b	2
d	1
e	2
f	2
g	3

Threshold = 3

Item	Head of node-link
a	
b	
d	
e	
f	
g	



Cond. FP-tree on "g" 3

{ (a:1, b:1, d:1, e:1, f:1, g:1),
(a:1, b:1, e:1, g:1),
(a:1, f:1, g:1)}

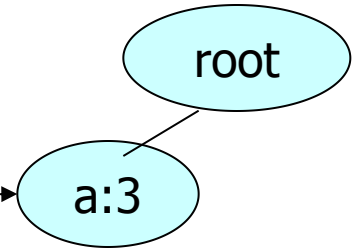
{(a:1, g:1),
(a:1, g:1),
(a:1, g:1) }

conditional pattern base of "g"

Item	Frequency
a	3
b	2
d	1
e	2
f	2
g	3

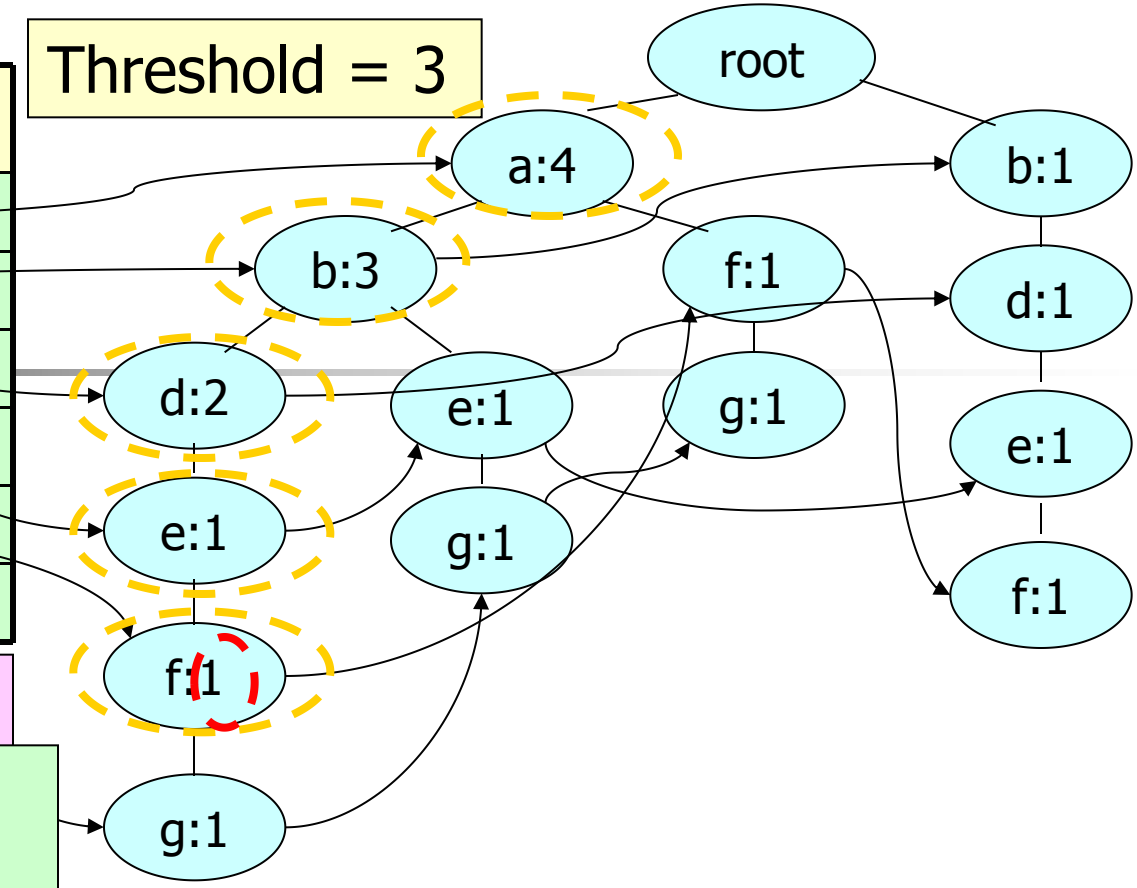
Item	Frequency
a	3
g	3

Item	Head of node-link
a	



Threshold = 3

Item	Head of node-link
a	
b	
d	
e	
f	
g	



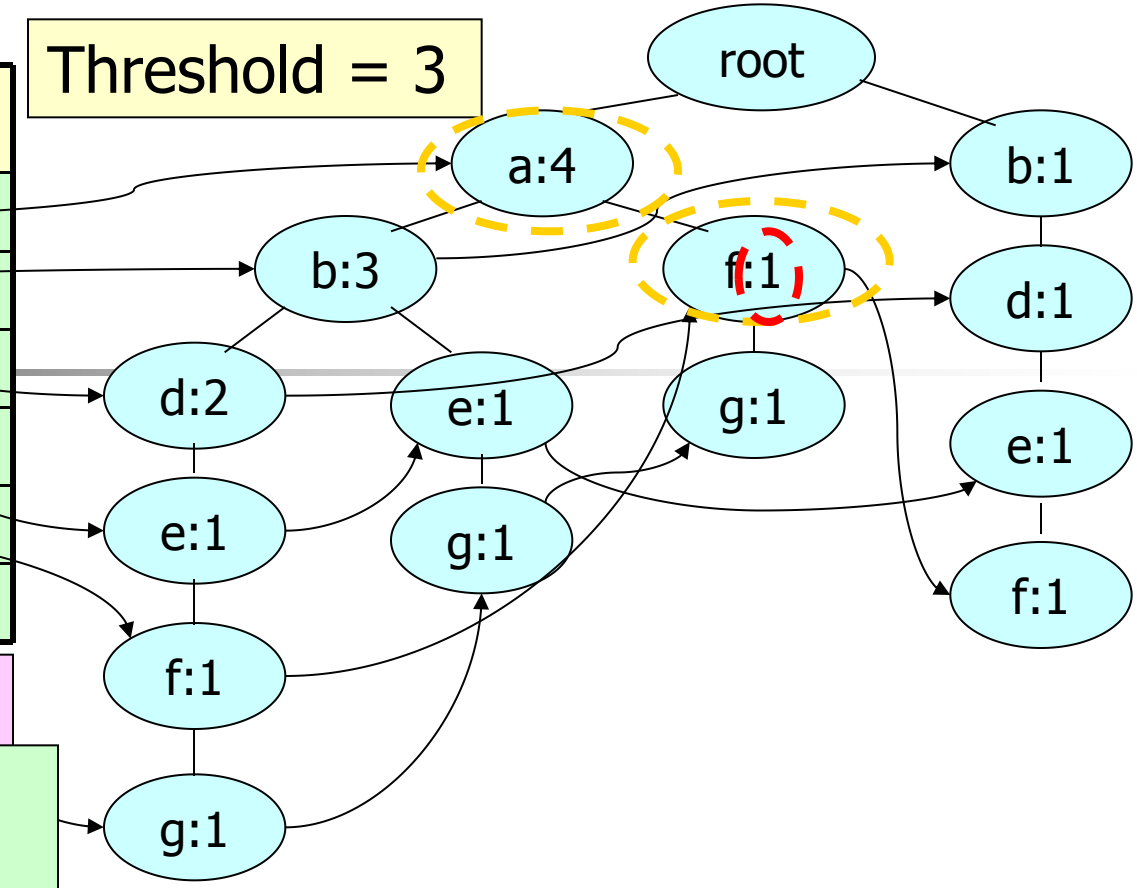
Cond. FP-tree on “f”

{ (a:1, b:1, d:1, e:1, f:1),

}

Threshold = 3

Item	Head of node-link
a	
b	
d	
e	
f	
g	

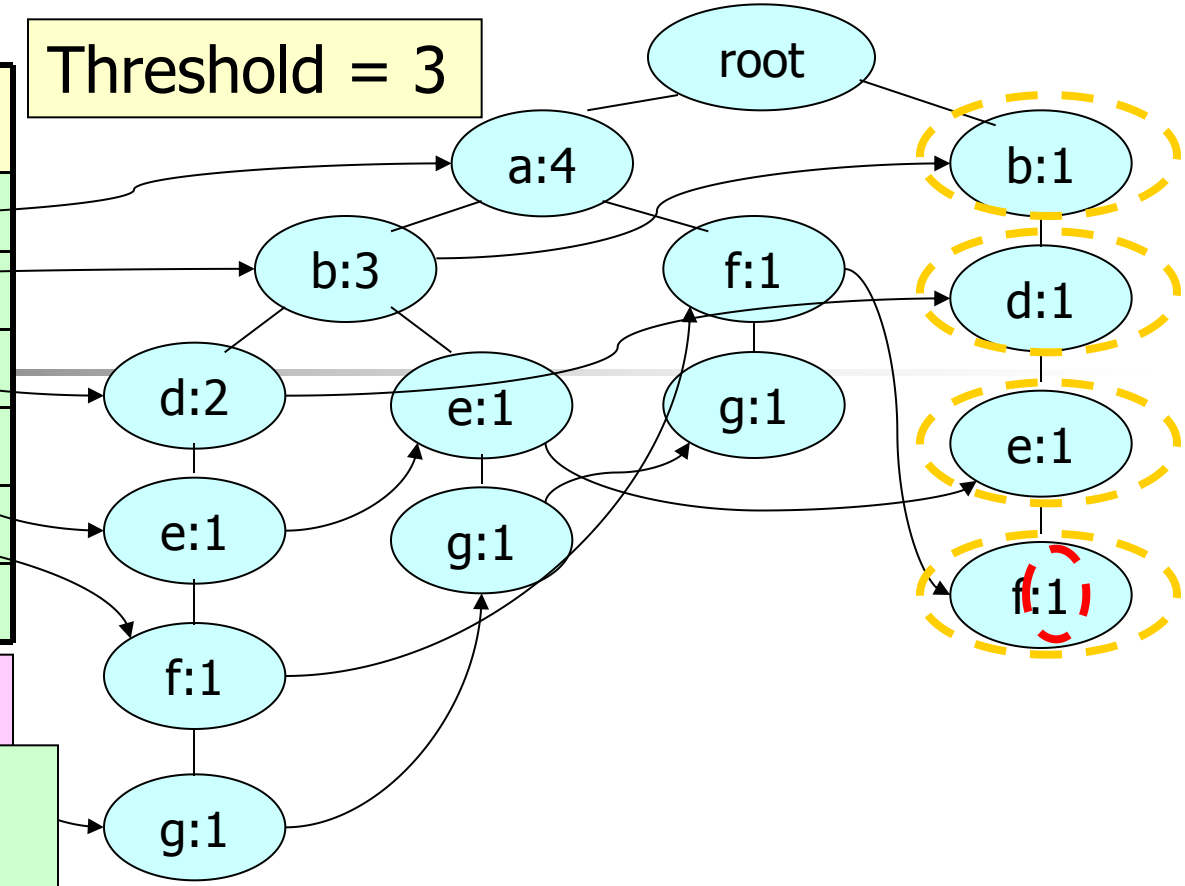


Cond. FP-tree on “f”

{ (a:1, b:1, d:1, e:1, f:1),
(a:1, f:1),
}

Threshold = 3

Item	Head of node-link
a	
b	
d	
e	
f	
g	



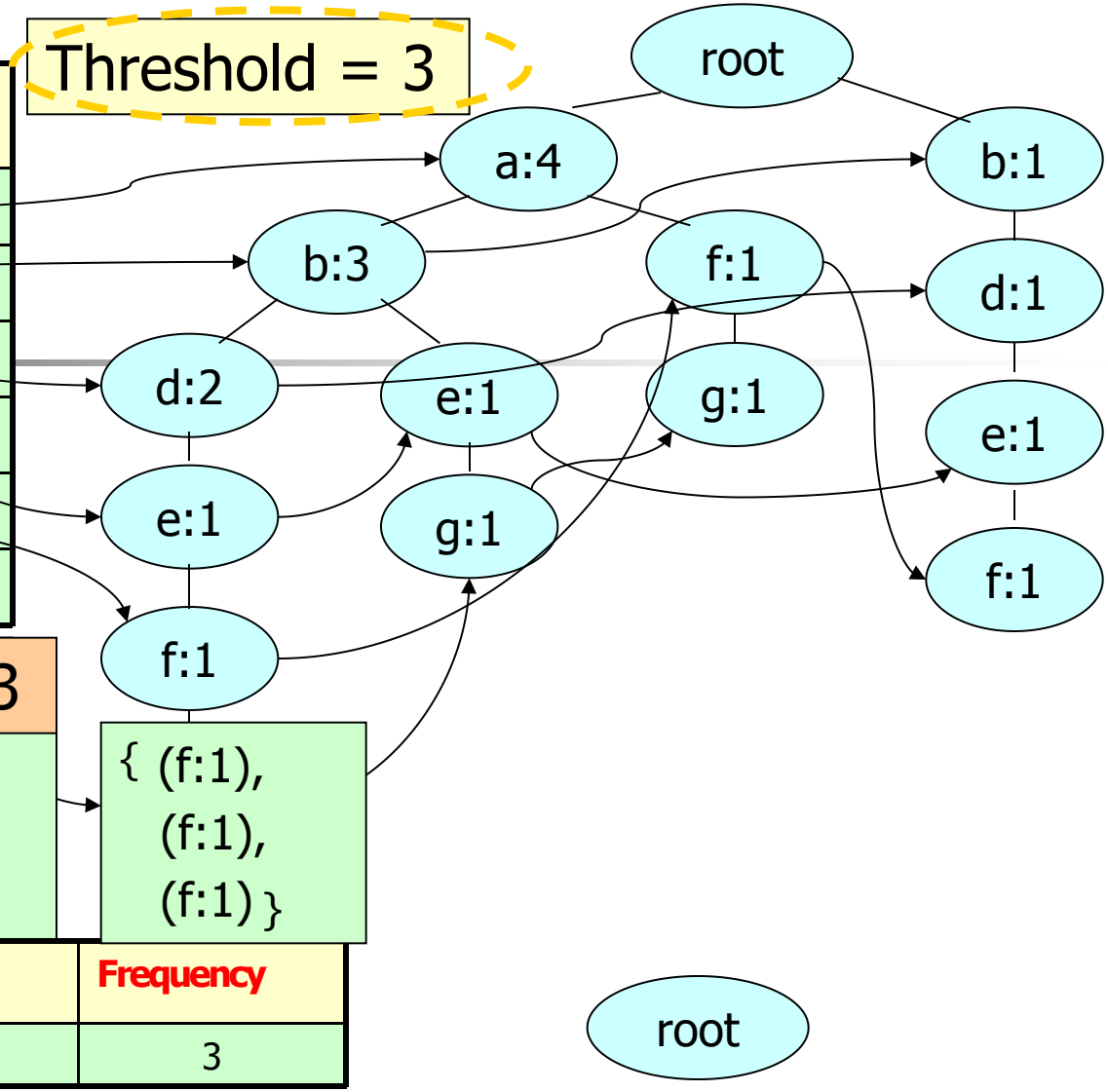
Cond. FP-tree on "f"

{ (a:1, b:1, d:1, e:1, f:1),
(a:1, f:1),
(b:1, d:1, e:1, f:1)}

Item	Frequency
a	2
b	2
d	2
e	2
f	3
g	0

Threshold = 3

Item	Head of node-link
a	
b	
d	
e	
f	
g	



Cond. FP-tree on "f" 3

{ (a:1, b:1, d:1, e:1, f:1),
(a:1, f:1),
(b:1, d:1, e:1, f:1)}

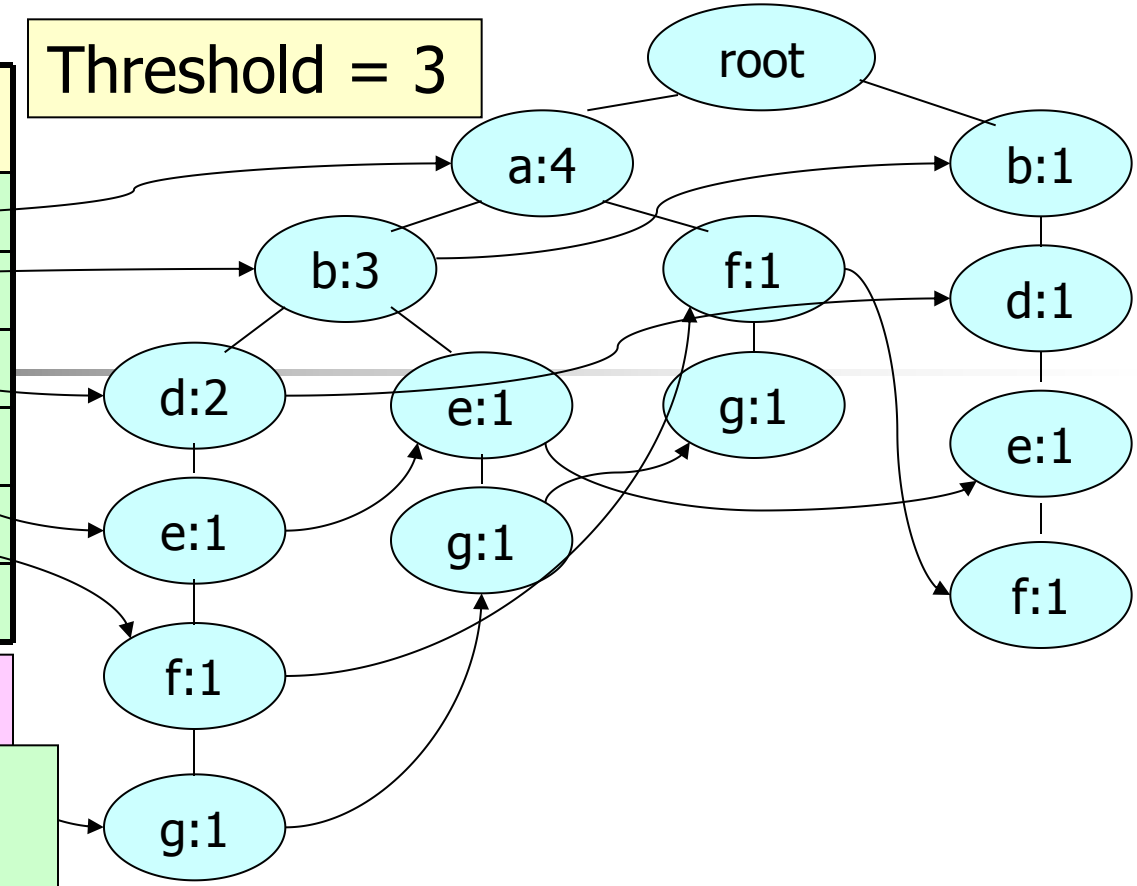
{ (f:1),
(f:1),
(f:1) }

Item	Frequency
a	2
b	2
d	2
e	2
f	3
g	0

Item	Frequency
f	3

Threshold = 3

Item	Head of node-link
a	
b	
d	
e	
f	
g	



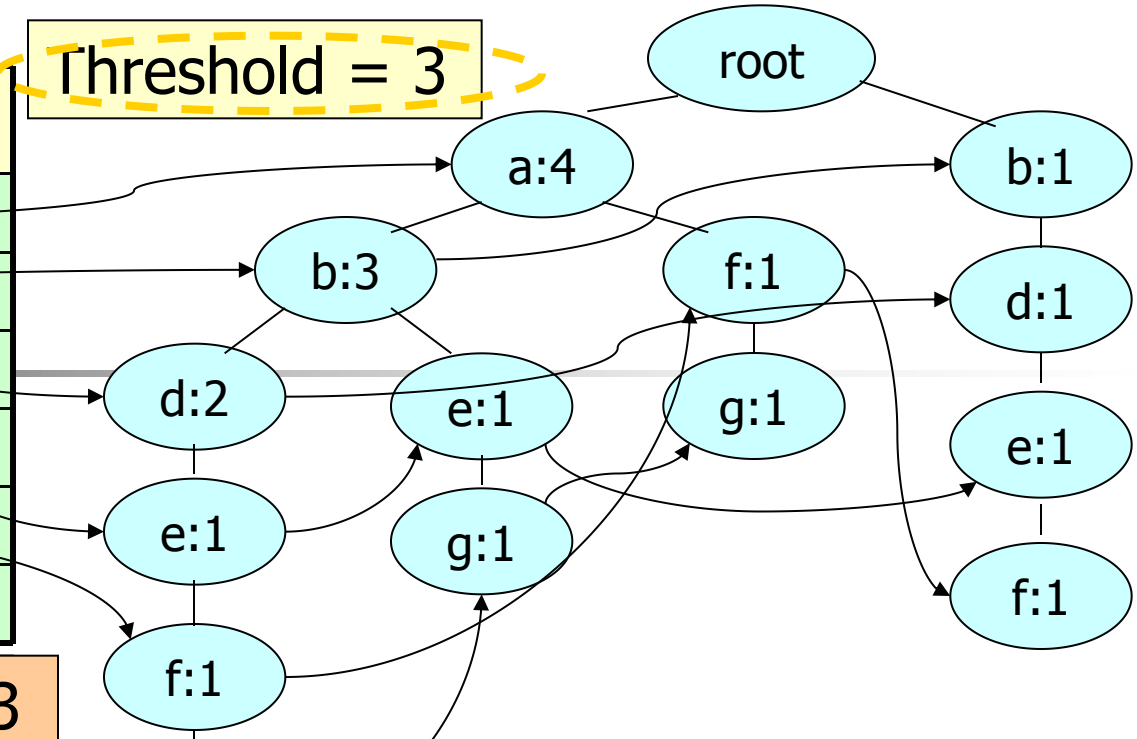
Cond. FP-tree on “e”

{ (a:1, b:1, d:1, e:1),
(a:1, b:1, e:1),
(b:1, d:1, e:1) }

Item	Frequency
a	2
b	3
d	2
e	3
f	0
g	0

Threshold = 3

Item	Head of node-link
a	
b	
d	
e	
f	
g	



Cond. FP-tree on "e" 3

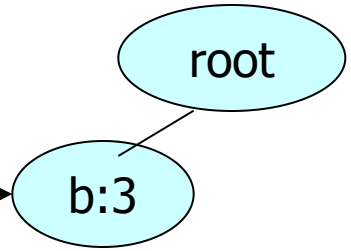
{ (a:1, b:1, d:1, e:1),
(a:1, b:1, e:1),
(b:1, d:1, e:1) }

{ (b:1, e:1),
(b:1, e:1),
(b:1, e:1) }

Item	Frequency
a	2
b	3
d	2
e	3
f	0
g	0

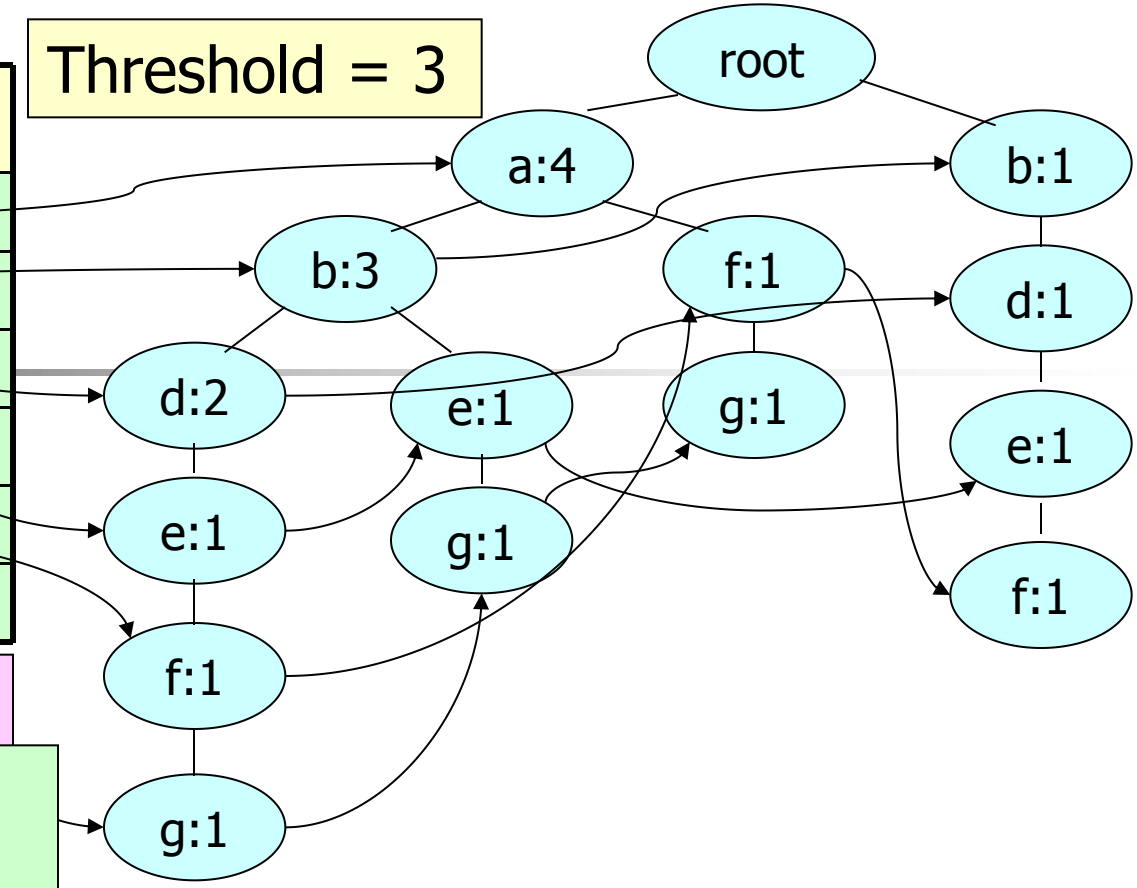
Item	Frequency
b	3
e	3

Item	Head of node-link
b	



Threshold = 3

Item	Head of node-link
a	
b	
d	
e	
f	
g	



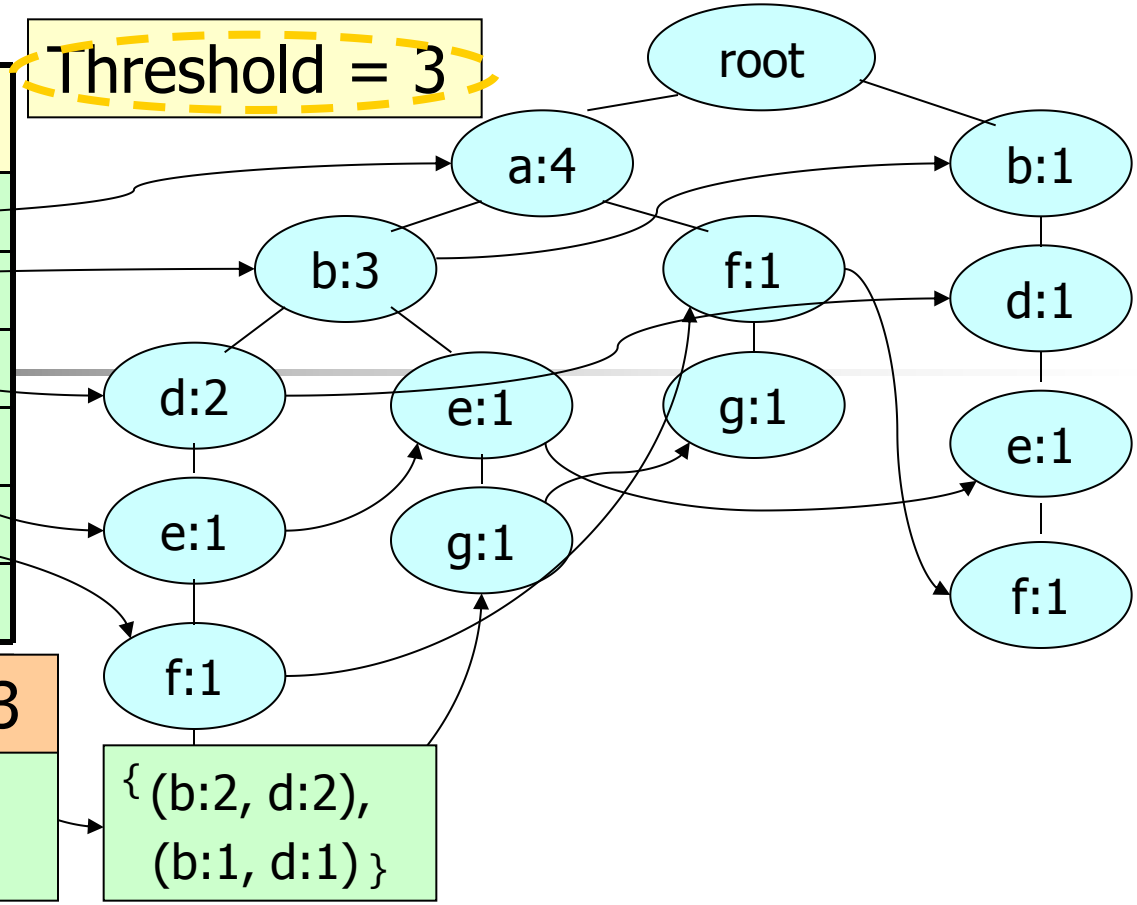
Cond. FP-tree on “d”

{ (a:2, b:2, d:2),
(b:1, d:1) }

Item	Frequency
a	2
b	3
d	3
e	0
f	0
g	0

Threshold = 3

Item	Head of node-link
a	
b	
d	
e	
f	
g	



Cond. FP-tree on "d" 3

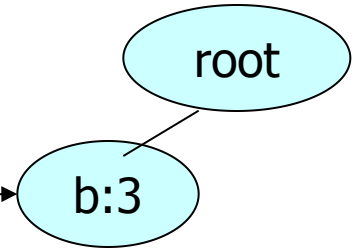
{ (a:2, b:2, d:2), (b:1, d:1) }

{ (b:2, d:2), (b:1, d:1) }

Item	Frequency
a	2
b	3
d	3
e	0
f	0
g	0

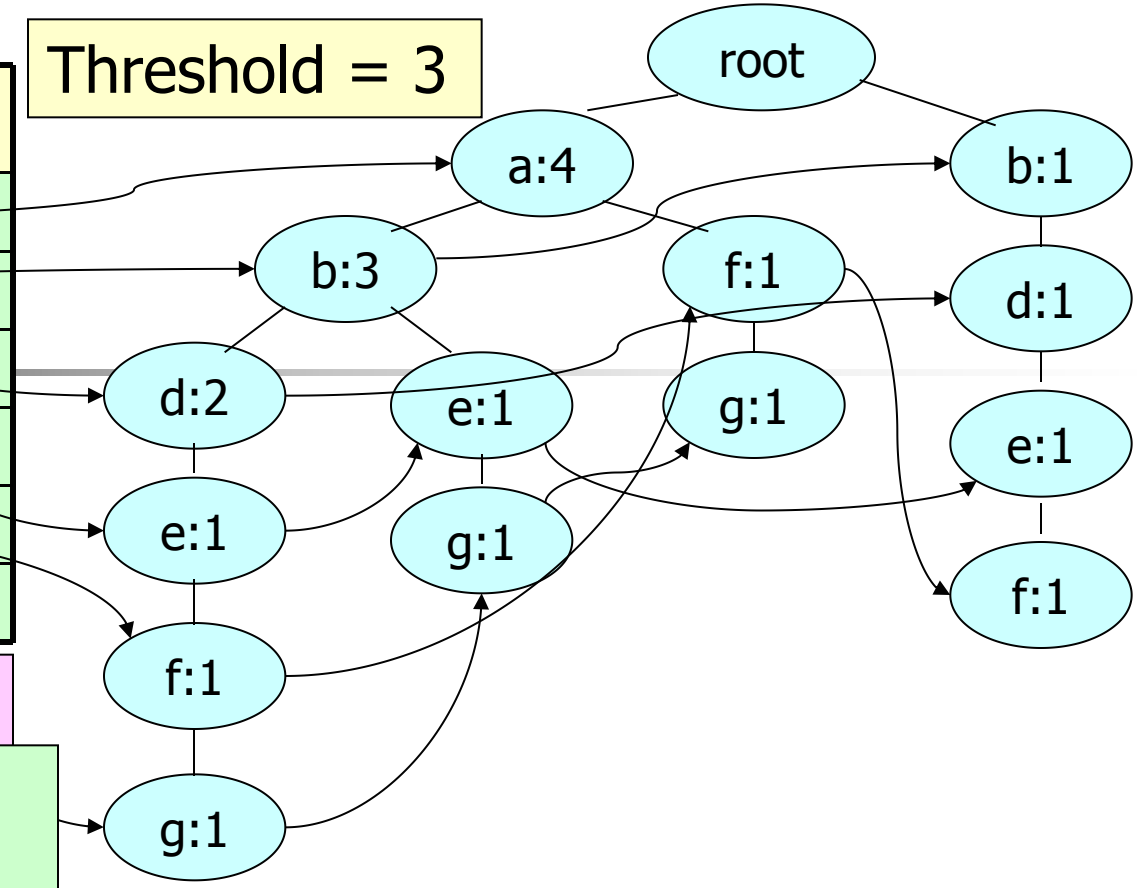
Item	Frequency
b	3
d	3

Item	Head of node-link
b	



Threshold = 3

Item	Head of node-link
a	
b	
d	
e	
f	
g	



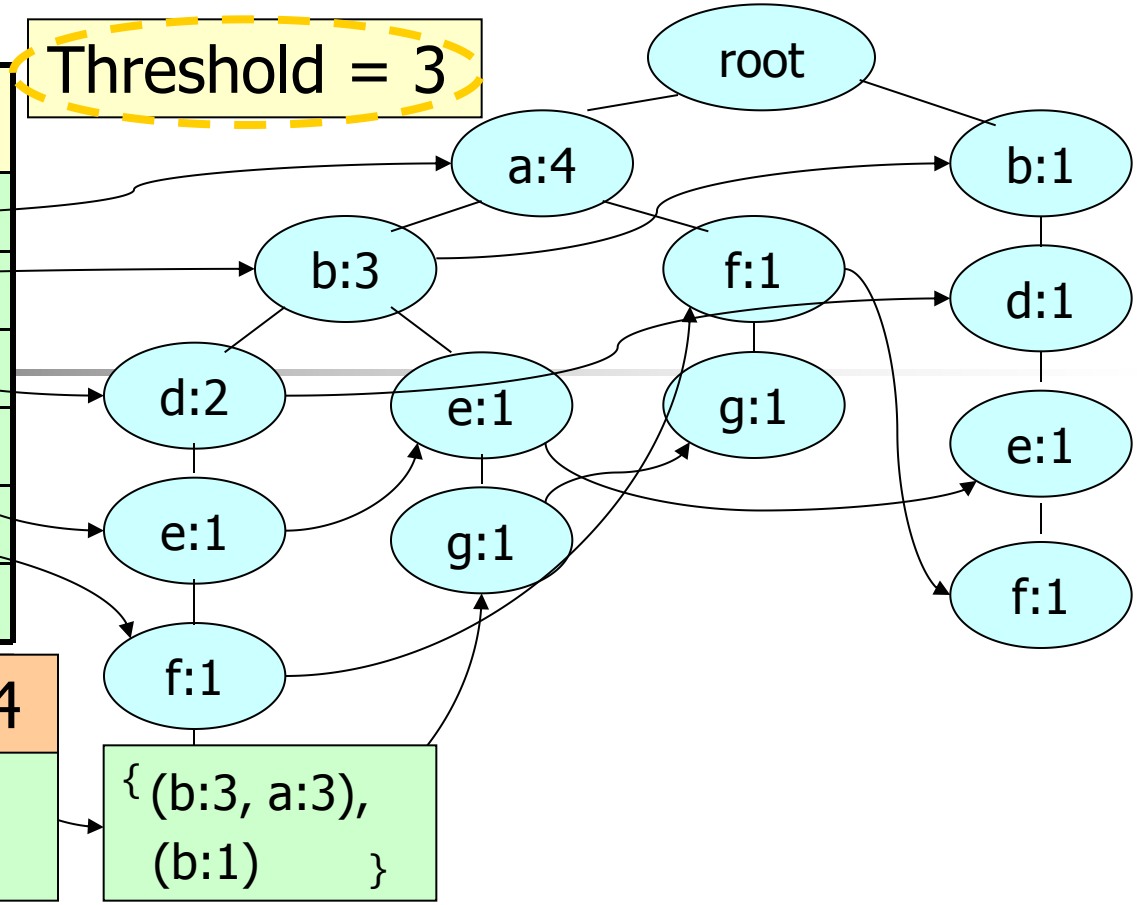
Cond. FP-tree on “b”

{ (a:3, b:3), (b:1) }

Item	Frequency
a	3
b	4
d	0
e	0
f	0
g	0

Threshold = 3

Item	Head of node-link
a	
b	
d	
e	
f	
g	



Cond. FP-tree on "b" 4

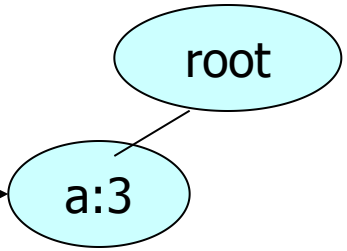
{ (a:3, b:3), (b:1) }

{ (b:3, a:3), (b:1) }

Item	Frequency
a	3
b	4
d	0
e	0
f	0
g	0

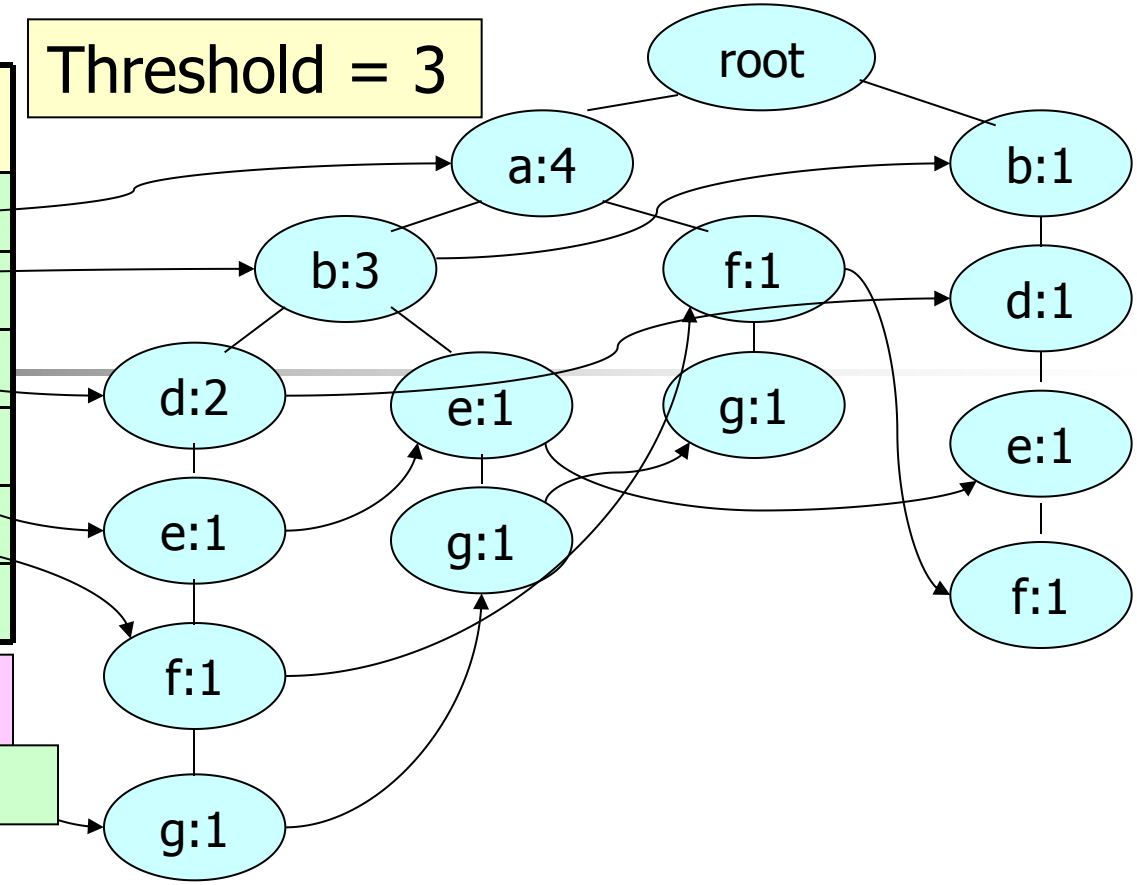
Item	Frequency
b	4
a	3

Item	Head of node-link
a	



Threshold = 3

Item	Head of node-link
a	
b	
d	
e	
f	
g	



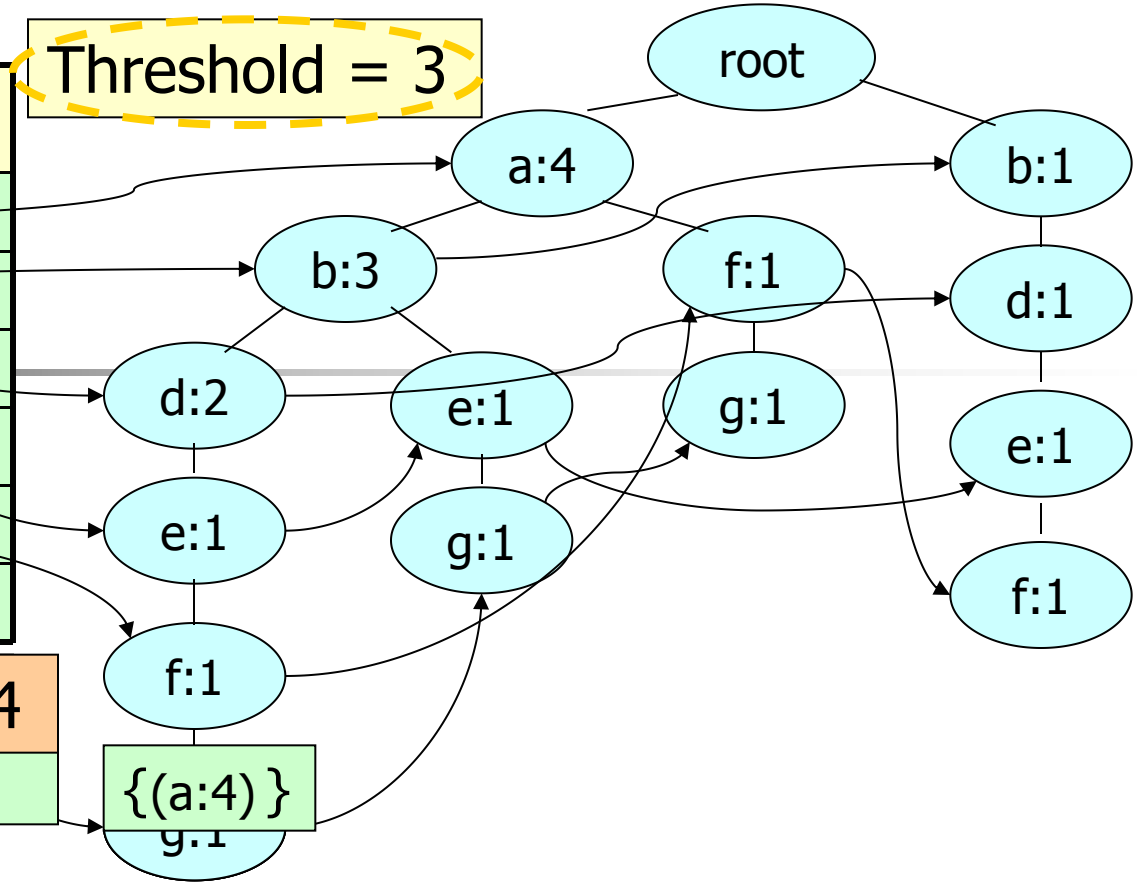
Cond. FP-tree on “a”

{ (a:4) }

Item	Frequency
a	4
b	0
d	0
e	0
f	0
g	0

Threshold = 3

Item	Head of node-link
a	
b	
d	
e	
f	
g	

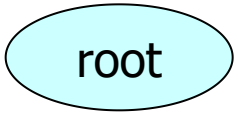


Cond. FP-tree on “a” 4

{ (a:4) }

Item	Frequency
a	4
b	0
d	0
e	0
f	0
g	0

Item	Frequency
a	4



FP-tree

Step 1: Deduce the ordered frequent items. For items with the same frequency, the order is given by the alphabetical order.

Step 2: Construct the FP-tree from the above data

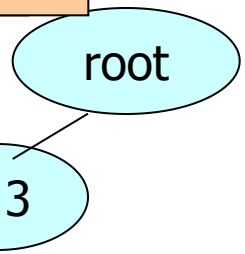
Step 3: From the FP-tree above, construct the FP-conditional tree for each item (or itemset).

Step 4: Determine the frequent patterns.

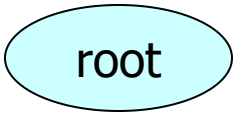
Cond. FP-tree on “g” 3

Cond. FP-tree on “g” 3

Item	Head of node-link
a	



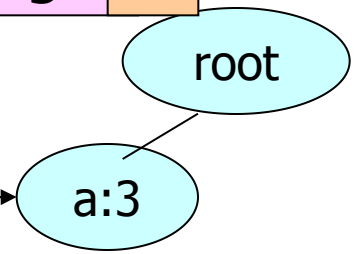
Cond. FP-tree on “f” 3



Cond. FP-tree on “e” 3

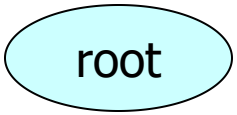
Cond. FP-tree on “g” 3

Item	Head of node-link
a	



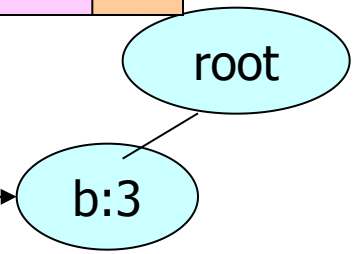
Cond. FP-tree on “d” 3

Cond. FP-tree on “f” 3



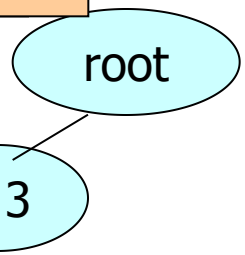
Cond. FP-tree on “e” 3

Item	Head of node-link
b	



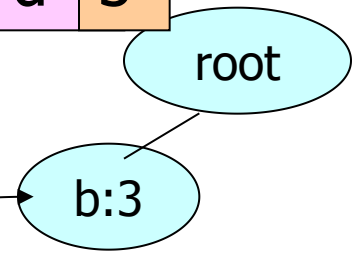
Cond. FP-tree on “g” 3

Item	Head of node-link
a	

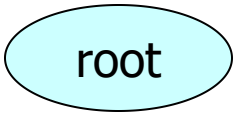


Cond. FP-tree on “d” 3

Item	Head of node-link
b	



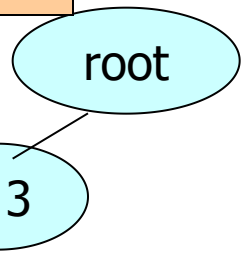
Cond. FP-tree on “f” 3



Cond. FP-tree on “b” 4

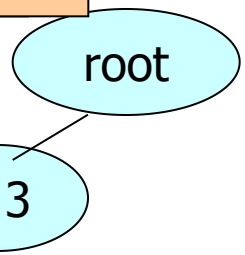
Cond. FP-tree on “e” 3

Item	Head of node-link
b	



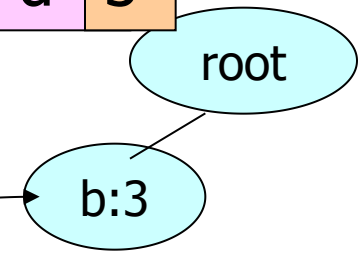
Cond. FP-tree on “g” 3

Item	Head of node-link
a	

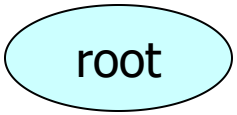


Cond. FP-tree on “d” 3

Item	Head of node-link
b	

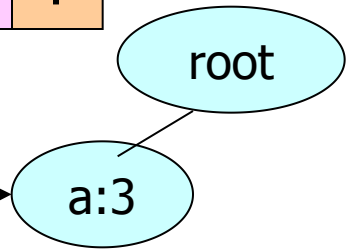


Cond. FP-tree on “f” 3



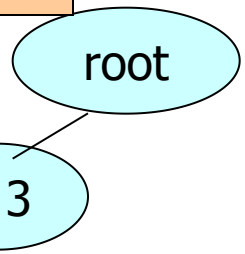
Cond. FP-tree on “b” 4

Item	Head of node-link
a	

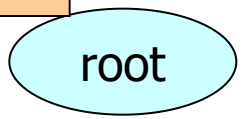


Cond. FP-tree on “e” 3

Item	Head of node-link
b	

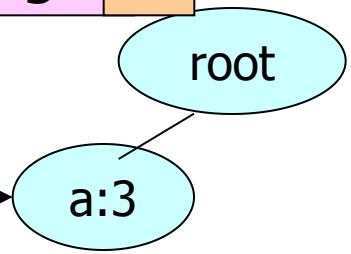


Cond. FP-tree on “a” 4



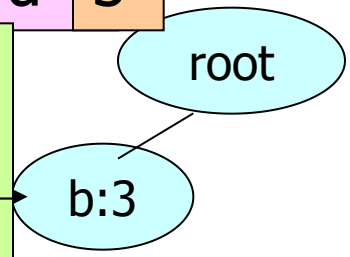
Cond. FP-tree on “g” 3

1. Before generating this cond. tree, we generate {g} (support = 3)
2. After generating this cond. tree, we generate {a, g} (support = 3)



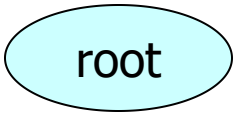
Cond. FP-tree on “d” 3

1. Before generating this cond. tree, we generate {d} (support = 3)
2. After generating this cond. tree, we generate {b, d} (support = 3)



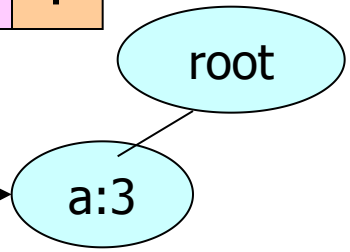
Cond. FP-tree on “f” 3

1. Before generating this cond. tree, we generate {f} (support = 3)
2. After generating this cond. tree, we we do not generate any itemset.



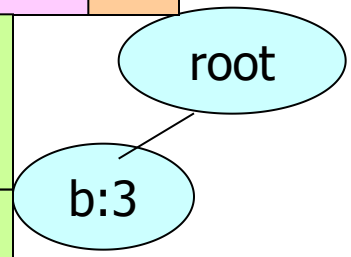
Cond. FP-tree on “b” 4

1. Before generating this cond. tree, we generate {b} (support = 4)
2. After generating this cond. tree, we generate {a, b} (support = 3)



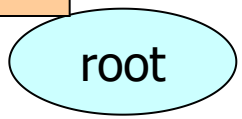
Cond. FP-tree on “e” 3

1. Before generating this cond. tree, we generate {e} (support = 3)
2. After generating this cond. tree, we generate {b, e} (support = 3)



Cond. FP-tree on “a” 4

1. Before generating this cond. tree, we generate {a} (support = 4)
2. After generating this cond. tree, we do not generate any itemset.



Complexity

- Complexity in building FP-tree
  - Two scans of the transactions DB
    - Collect frequent items
    - Construct the FP-tree
- Cost to insert one transaction
  - Number of frequent items in this transaction

Size of the FP-tree

- The size of the FP-tree is bounded by the overall occurrences of the frequent items in the database

Height of the Tree

- The height of the tree is bounded by the maximum number of frequent items in any transaction in the database

Compression

- With respect to the total number of items stored,
  - is FP-tree more compressed compared with the original databases?

Details of the Algorithm

- Procedure FP-growth (Tree, α)
  - if Tree contains a single path P
    - for each combination (denoted by β) of the nodes in the path P do
      - generate pattern $\beta \cup \alpha$ with support = minimum support of nodes in β
  - else
    - for each a_i in the header table of Tree do
      - generate pattern $\beta = a_i \cup \alpha$ with support = a_i.support
      - construct β's conditional pattern base and then β's conditional FP-tree $Tree_\beta$
      - if $Tree_\beta \neq \varnothing$
        - Call FP-growth($Tree_\beta$, β)