

Association Rules. FP-tree

Data mining lab 7. Part II

Association rules generation

- Step 1. Find all frequent itemsets F_i , $2 \leq i \leq T$,
T -total number of items
- Step 2. Generate rules from the frequent itemsets

Tutorial exercise 2. FP-tree

Order ID	A	B	C	D	E	F
1	1	1			1	
2			1	1		
3		1				1
4	1			1		
5	1	1		1	1	
6	1	1		1		

1. Count items, put counts into a header table in descending order

A	4
B	4
C	1
D	4
E	2
F	1
F	1



Item	Count
A	4
B	4
D	4
E	2
C	1
F	1



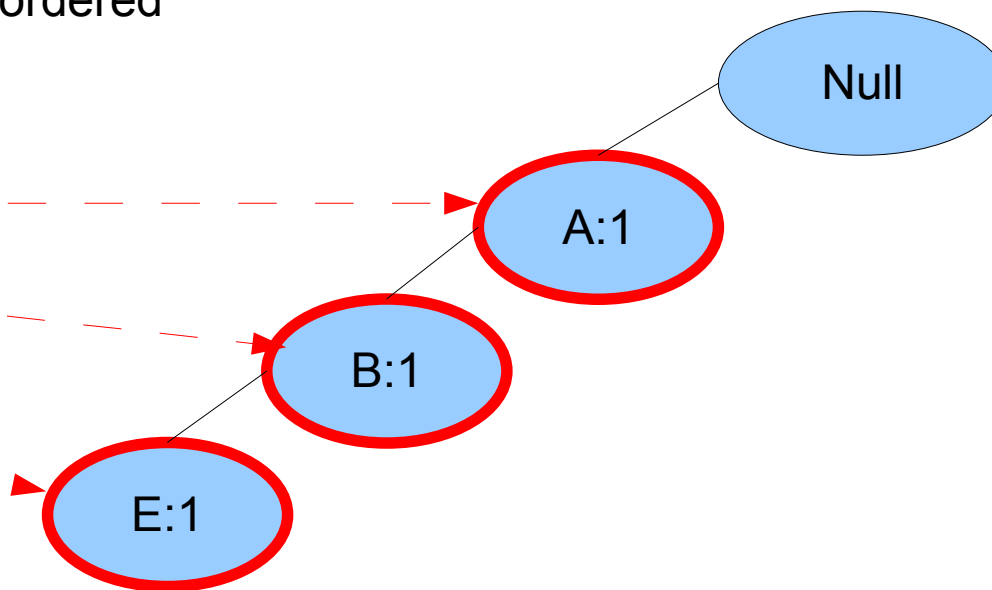
Item	Count
A	4
B	4
D	4
E	2

1. Insert transaction 1 into FP-tree

Order ID	A	B	C	D	E	F
1	1	1			1	
2			1	1		
3		1				1
4	1			1		
5	1	1		1	1	
6	1	1		1		

Insert transaction items ordered according to a header

Item	Count
A	4
B	4
D	4
E	2



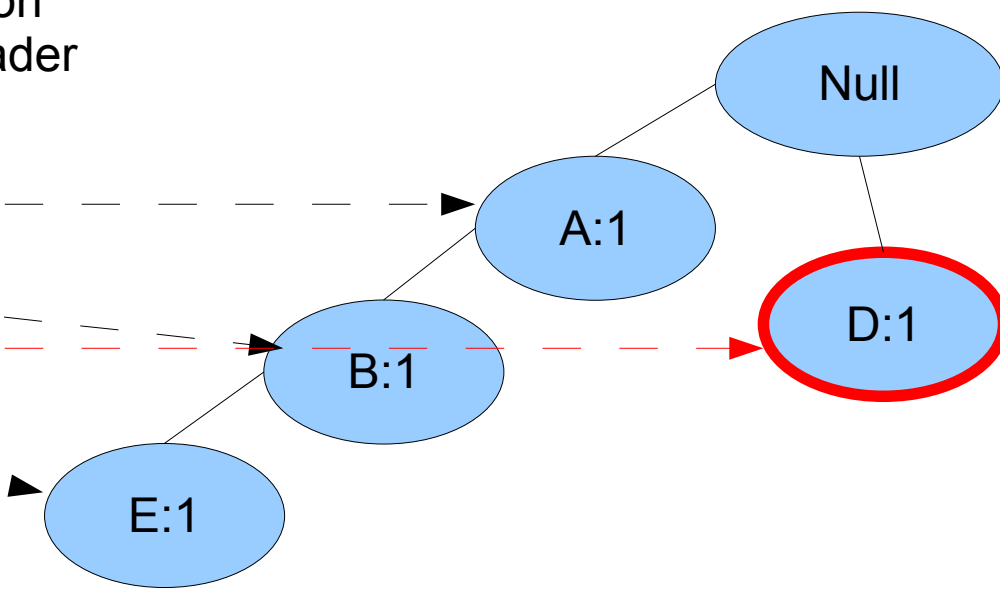
	A	B	D
B	1		
D			
E	1	1	

2. Insert transaction 2 into FP-tree

Order ID	A	B	C	D	E	F
1	1	1			1	
2			1	1		
3		1				1
4	1			1		
5	1	1		1	1	
6	1	1		1		

Insert ordering transaction items according to a header

Item	Count
A	4
B	4
D	4
E	2



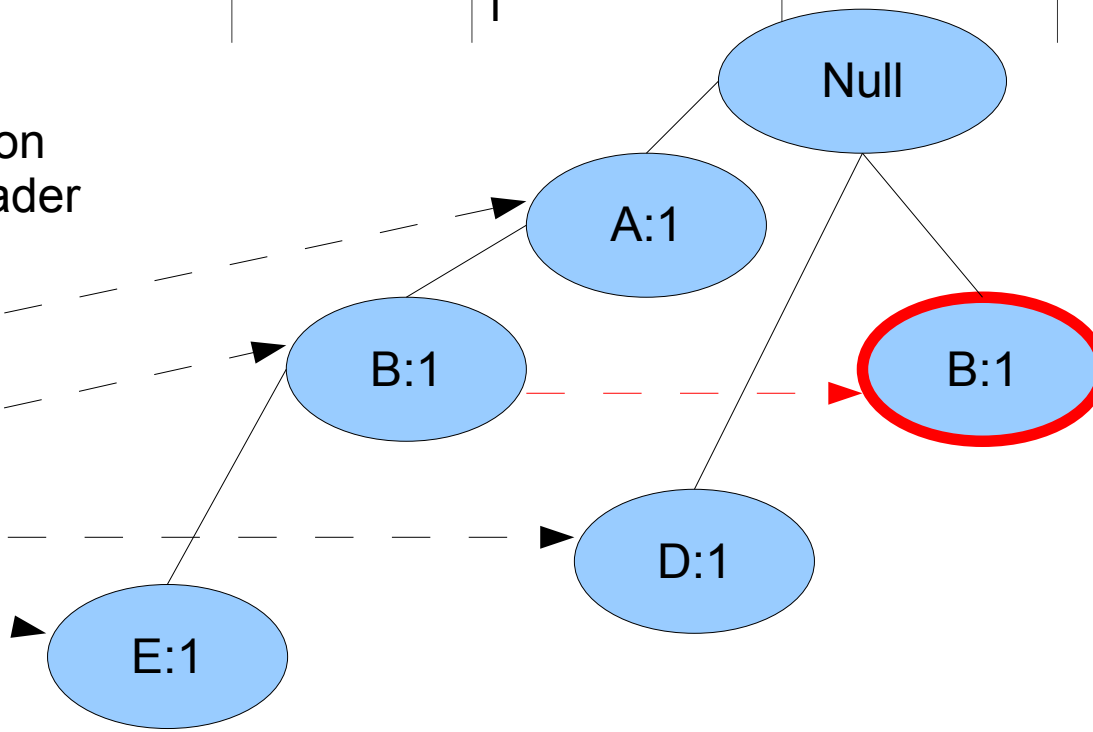
	A	B	D
B	1		
D			
E	1	1	

3. Insert transaction 3 into FP-tree

Order ID	A	B	C	D	E	F
1	1	1			1	
2			1	1		
3		1				1
4	1			1		
5	1	1		1	1	
6	1	1		1		

Insert ordering transaction items according to a header

Item	Count
A	4
B	4
D	4
E	2



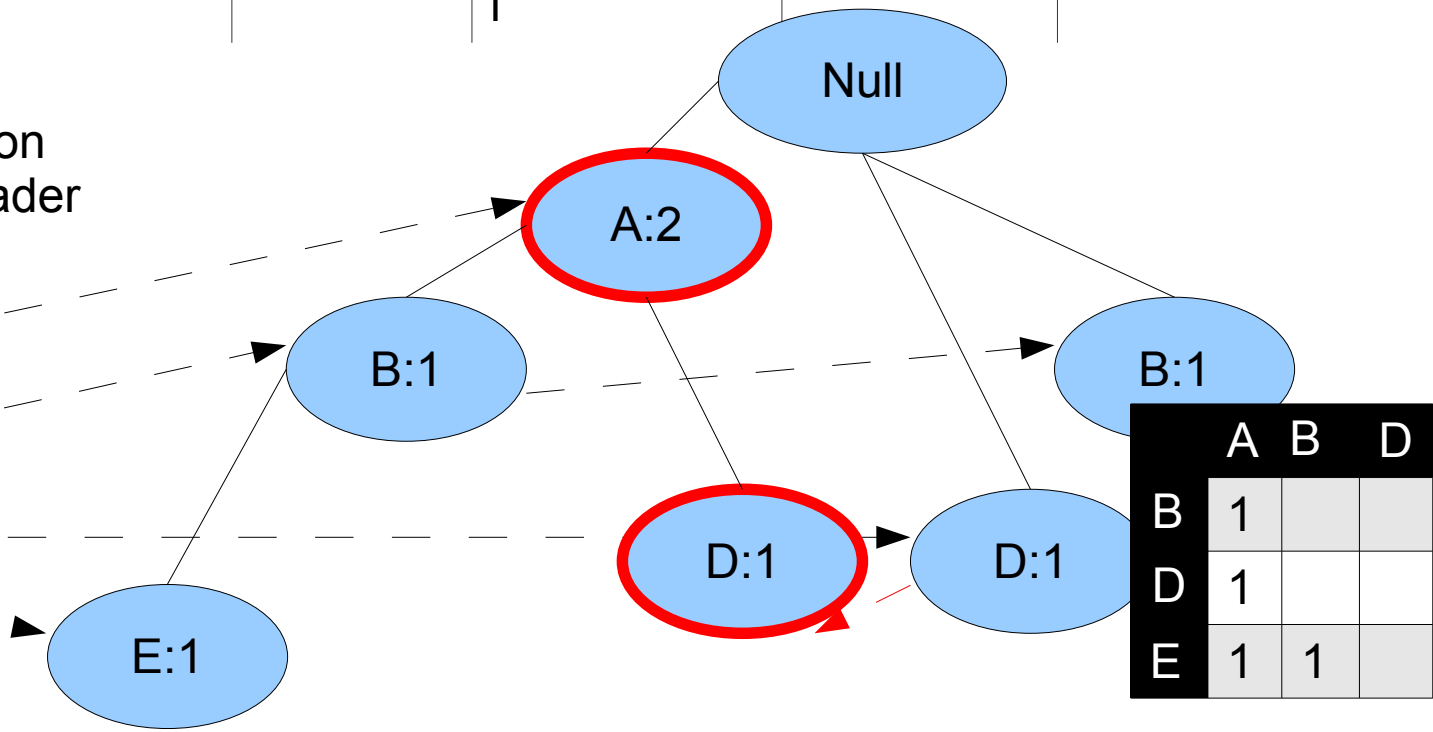
	A	B	D
B	1		
D			
E	1	1	

4. Insert transaction 4 into FP-tree

Order ID	A	B	C	D	E	F
1	1	1			1	
2			1	1		
3		1				1
4	1			1		
5	1	1		1	1	
6	1	1		1		

Insert ordering transaction items according to a header

Item	Count
A	4
B	4
D	4
E	2



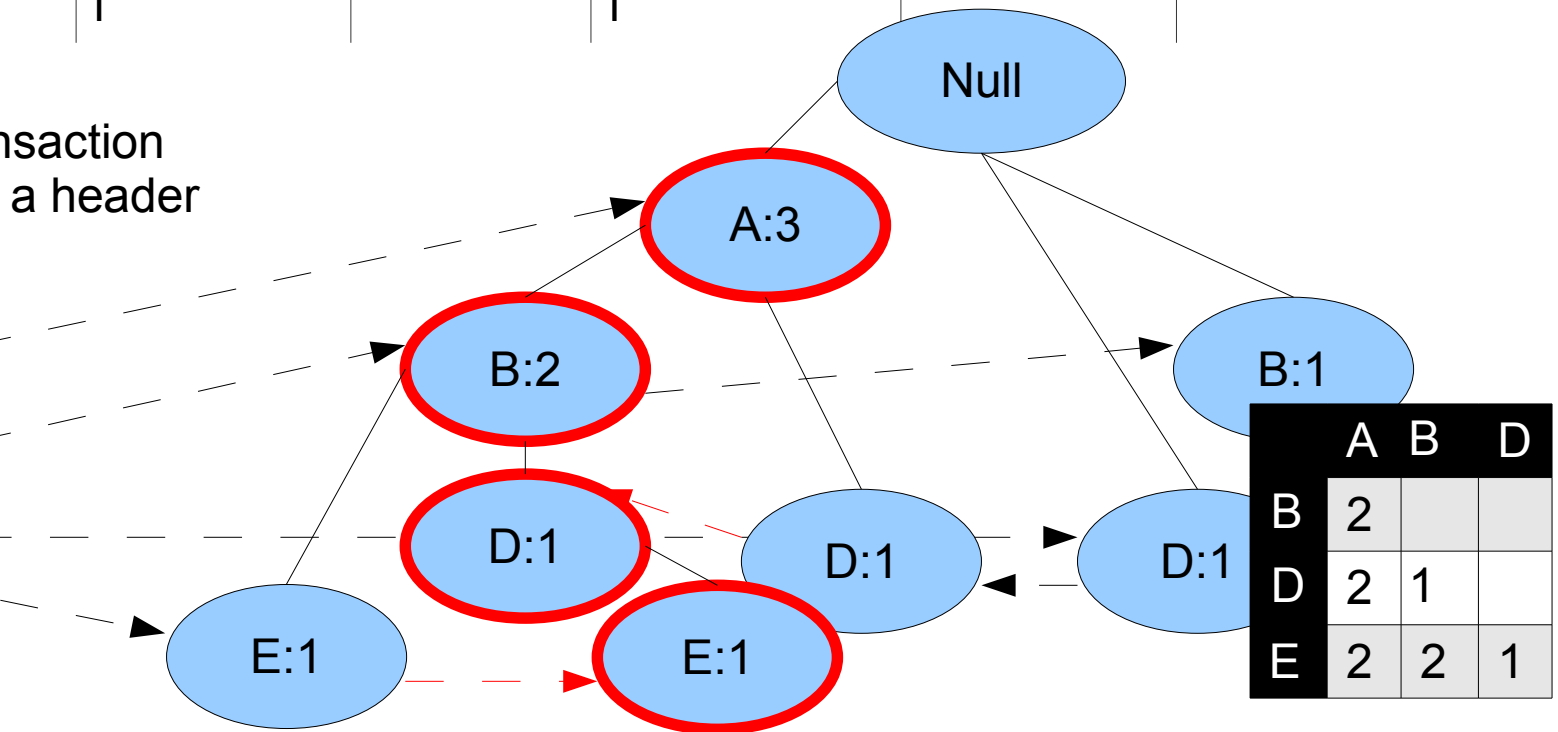
	A	B	D
B	1		
D	1		
E	1	1	

5. Insert transaction 5 into FP-tree

Order ID	A	B	C	D	E	F
1	1	1			1	
2			1	1		
3		1				1
4	1			1		
5	1	1		1	1	
6	1	1		1		

Insert ordering transaction items according to a header

Item	Count
A	4
B	4
D	4
E	2

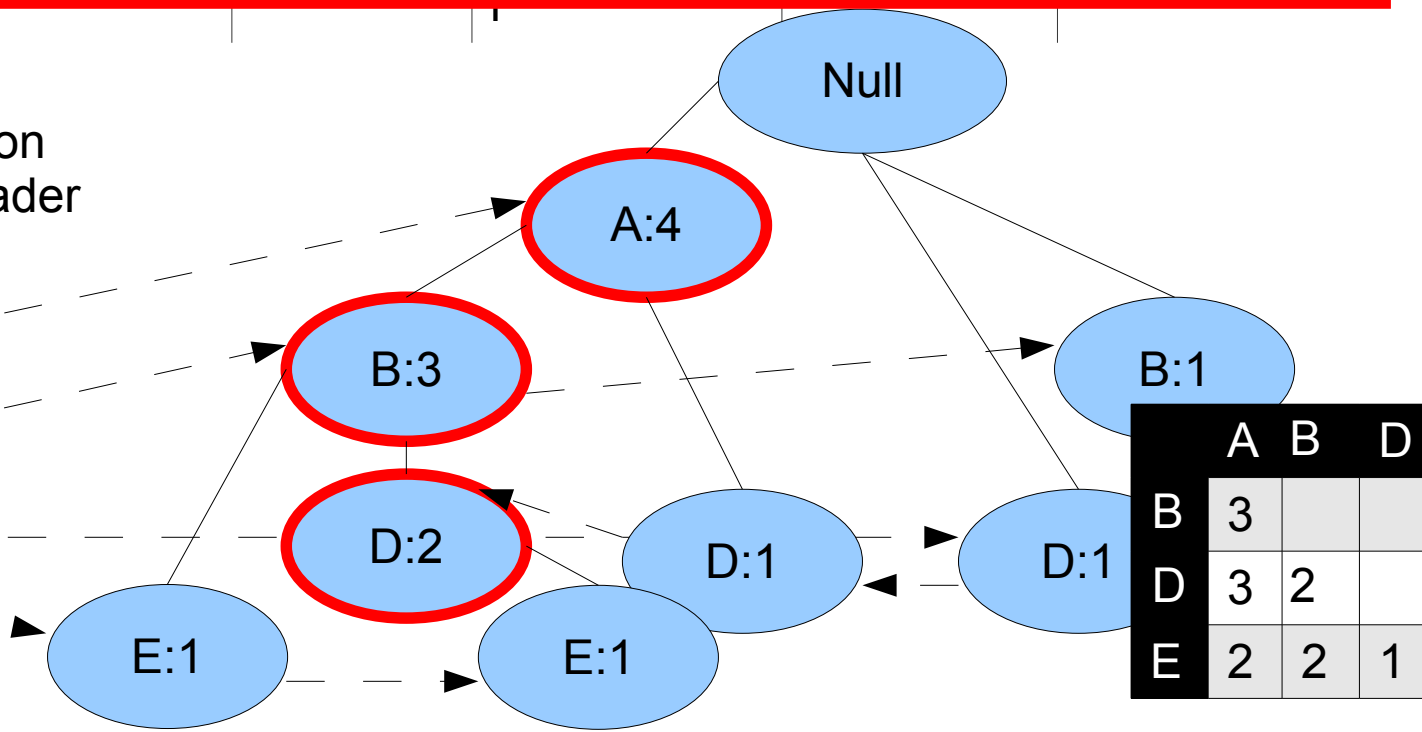


6. Insert transaction 6 into FP-tree

Order ID	A	B	C	D	E	F
1	1	1			1	
2			1	1		
3		1				1
4	1			1		
5	1	1		1	1	
6	1	1		1		

Insert ordering transaction items according to a header

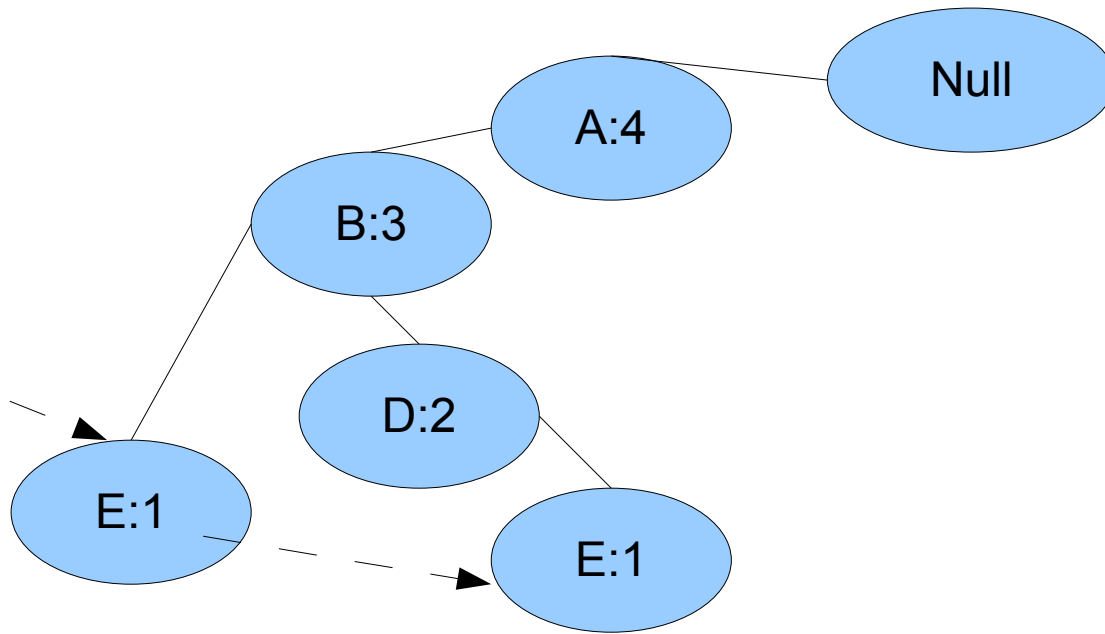
Item	Count
A	4
B	4
D	4
E	2



	A	B	D
B	3		
D	3	2	
E	2	2	1

7. Mining FP-tree

Item
A
B
D
E



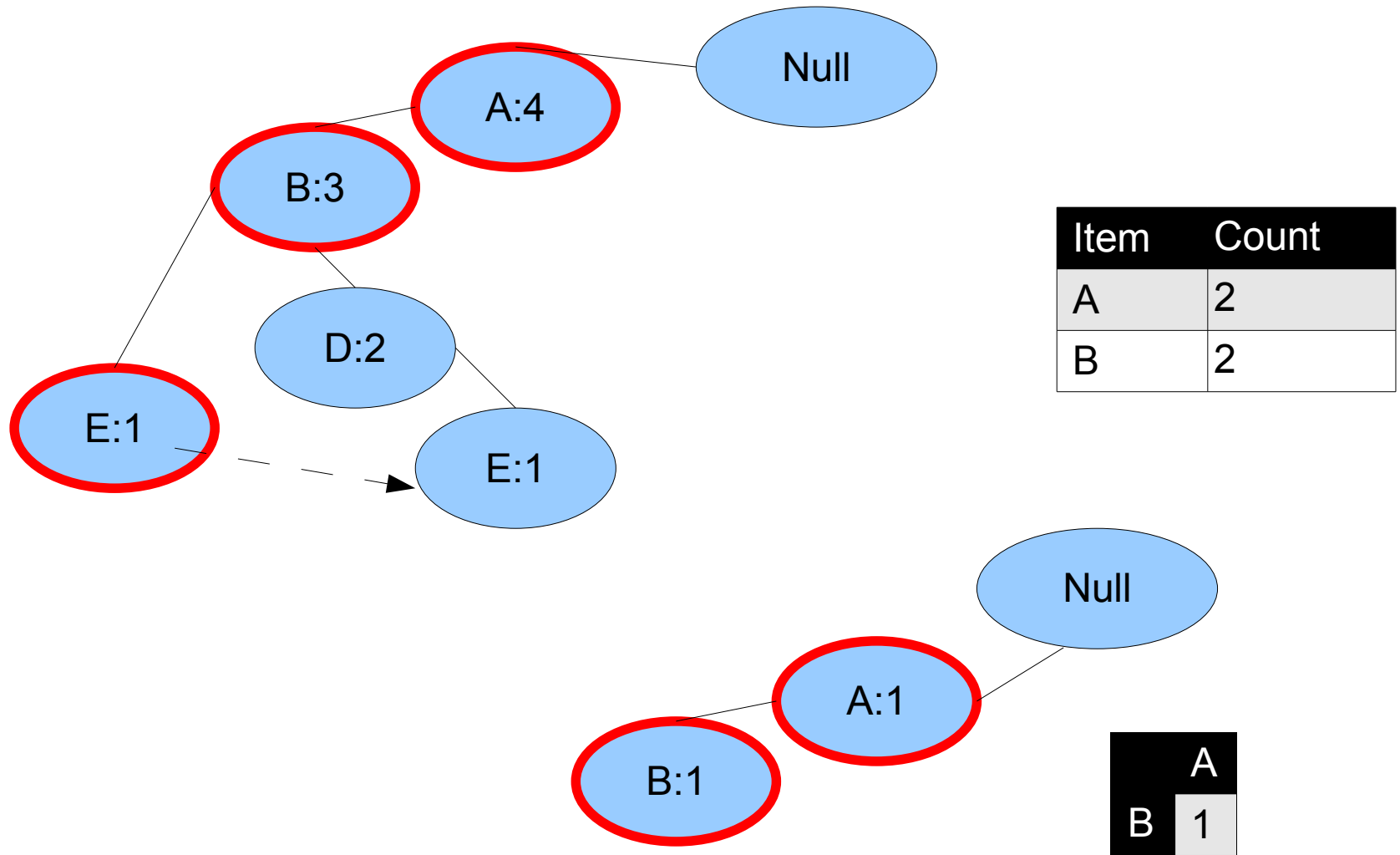
New header table

Item	Count
A	2
B	2
D	1

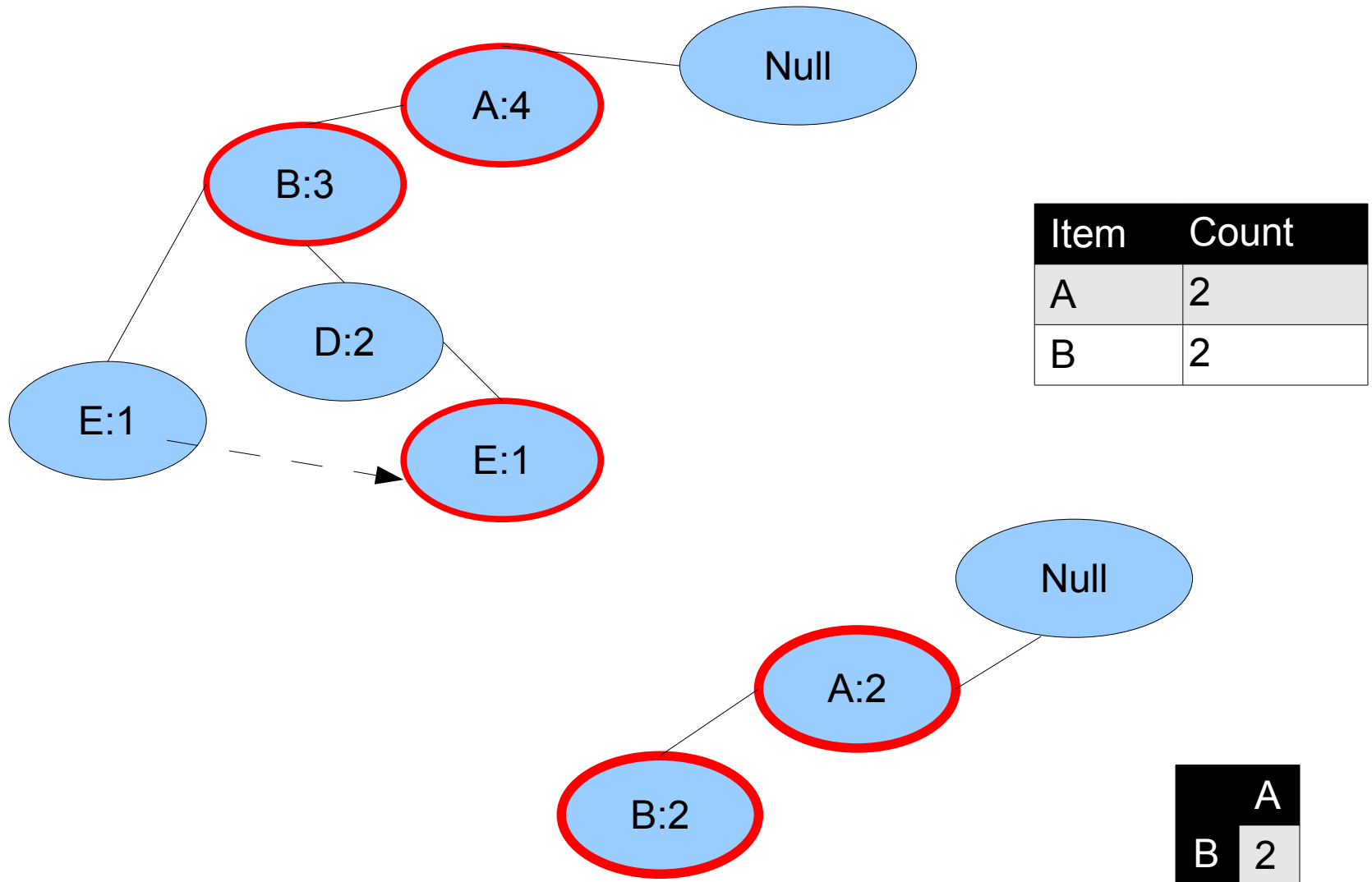


	A	B	D
B	3		
D	3	2	
E	2	2	1

7. All sets ending in E. Insert transactions into FP-tree conditional on E

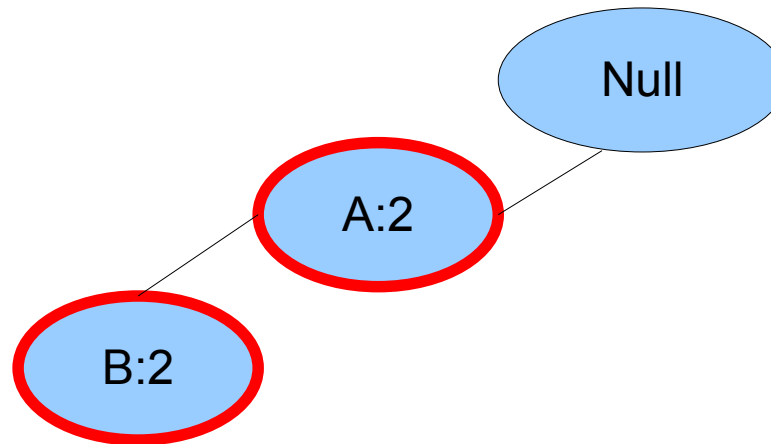


7. All sets ending in E. Insert transactions into FP-tree conditional on E



7. All sets ending in E. Add to the solution set

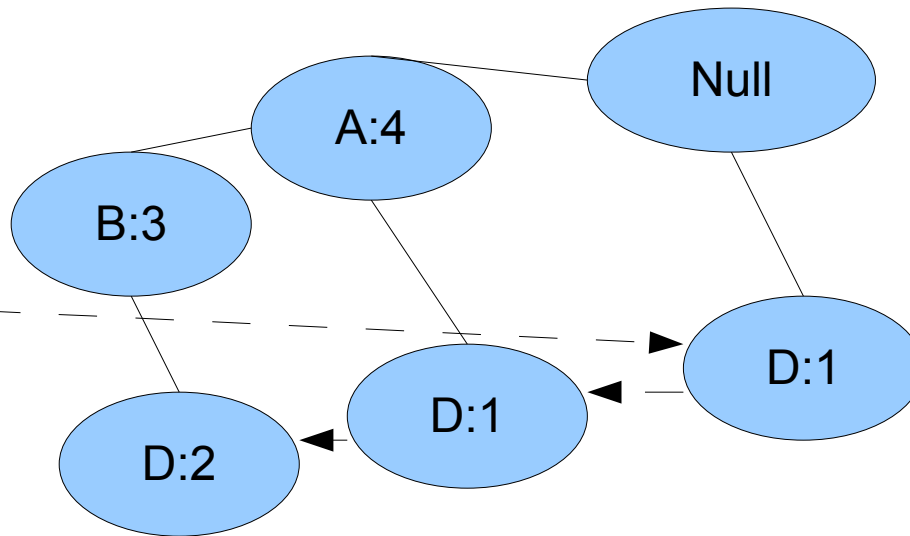
End of recursion. Single path



E-set={{A,E},{B,E},{A,B,E}}

8. Mining FP-tree

Item
A
B
D
E



New header table

Item	Count
A	3
B	2

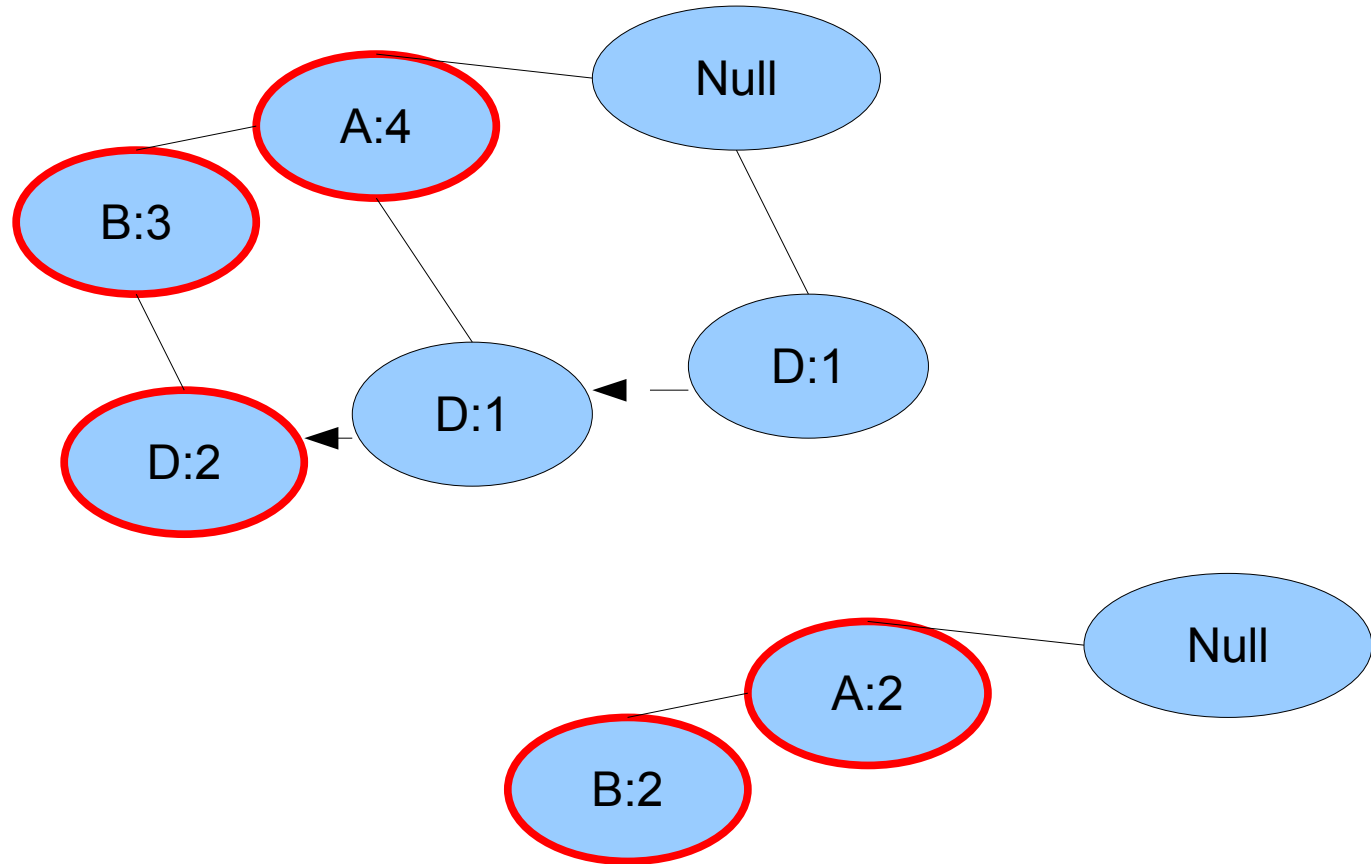


	A	B
B	3	
D	3	2

8. All sets ending in D. Add transactions to FP-tree conditional on D

New header table

Item	Count
A	3
B	2

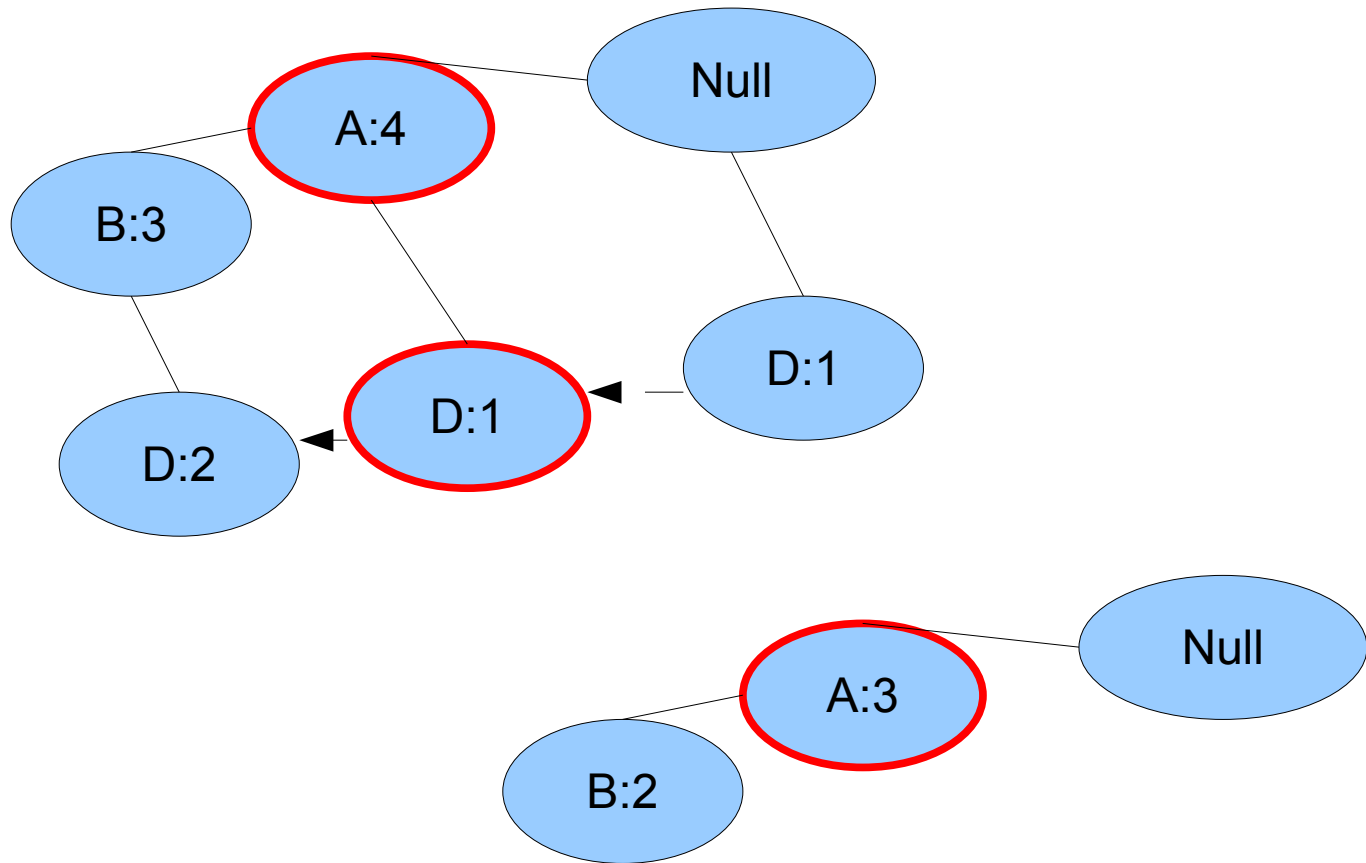


A
B 2

8. All sets ending in D. Add transactions to FP-tree conditional on D

New header table

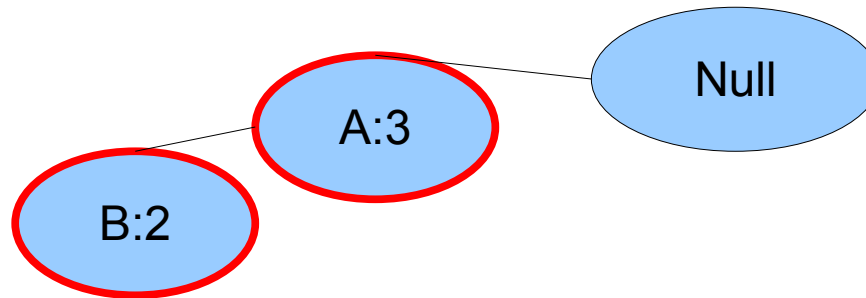
Item	Count
A	3
B	2



A
B 2

8. All sets ending in D. Add to the solution set

End of recursion. Single path



D-sets={{A,D},{B,D},{A,B,D}}

9. All sets ending in B

Similarly, we find:

B-sets={A,B}

10. Solution is the union of E,D,B-sets

E-set={{A,E},{B,E},{A,B,E}}

D-sets={{A,D},{B,D},{A,B,D}}

B-sets={A,B}

Solution: {{A,B}, {A,D}, {A,E}, {B,D}, {B,E}, {A,B,D}, {A,B,E}}

Which is exactly the same as computed by the Apriori algorithm:

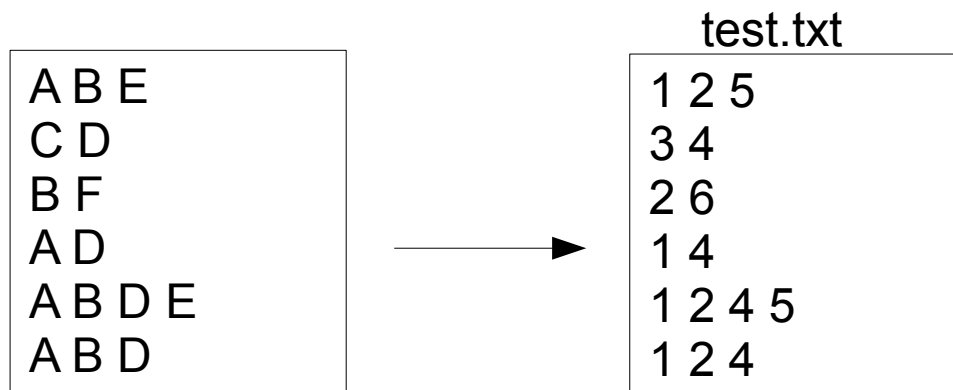
F2={{A,B},{A,D},{A,E},{B,D},{B,E}}

F3={{A,B,D},{A,B,E}}

11. Hints for the assignment 2.5.

FMI

- The input for the program is a file where each transaction is represented as a list of numeric values – column numbers. For example, for our pizza toppings example the input is:



Compile and run the program as usual:

Make

```
./fim_all test.txt 2 test_res.txt
```