FP Tree

- It eliminates both the drawbacks of Apriori algo.

- It is used to find frequent itemsets L_k without generating candidate sets C_k and it requires only two database scans.

- This method x'forms the database of x'actions into a tree structure called FP (frequent pattern) tree such that association information bet^n the data items is preserved

- The method has following steps.

(i) Scan the db once & find set of frequent 1-item sets L_1
(ii) Arrange the frequent items in descending order of support count.
(iii) Scan the db again & construct FP tree
(iv) For each node in FP tree, construct conditional pattern base.
(v) For each conditional pattern base construct conditional FP tree
(vi) Mine conditional FP trees recursively to grow frequent patterns.

Consider the following set of x'actions.

Tid	items
T_1	i_1 i_2 i_5
T_2	i_2 i_4
T_3	i_2 i_3
T_4	i_1 i_2 i_4
T_5	i_1 i_3
T_6	i_2 i_3
T_7	i_1 i_3
T_8	i_1 i_2 i_3 i_5
T_9	i_1 i_2 i_3

let min_support = 2

step 1 : Scan the db & find L_1

C_1 =	{ $\{i_1\}$	$\{i_2\}$	$\{i_3\}$	$\{i_4\}$	$\{i_5\}$ }
SC =	6	7	6	2	2

$\therefore$ L_1 = { $\{i_1\}$ $\{i_2\}$ $\{i_3\}$ $\{i_4\}$ $\{i_5\}$ }

step 2 : Arrange the frequent items in descending order of SC

$\therefore$ $i_2 \rightarrow 7$, $i_1 \rightarrow 6$, $i_3 \rightarrow 6$, $i_4 \rightarrow 2$, $i_5 \rightarrow 2$

Now the frequent items in each x'action must be processed in this order only, For this add a column "ordered frequent items" in the set of x'actions.

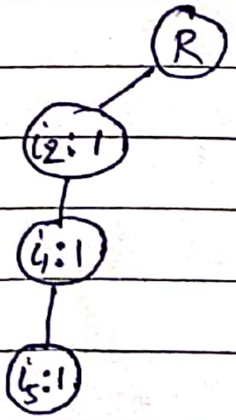
Tid	items	ordered frequent items
T1	i_1 i_2 i_5	i_2 i_1 i_5
T2	i_2 i_4	i_2 i_4
T3	i_2 i_3	i_2 i_3
T4	i_1 i_2 i_4	i_2 i_1 i_4
T5	i_1 i_3	i_1 i_3
T6	i_2 i_3	i_2 i_3
T7	i_1 i_3	i_1 i_3
T8	i_1 i_2 i_3 i_5	i_2 i_1 i_3 i_5
T9	i_1 i_2 i_3	i_2 i_1 i_3

step III Scan the DB again (the 3rd col) & construct FP tree.

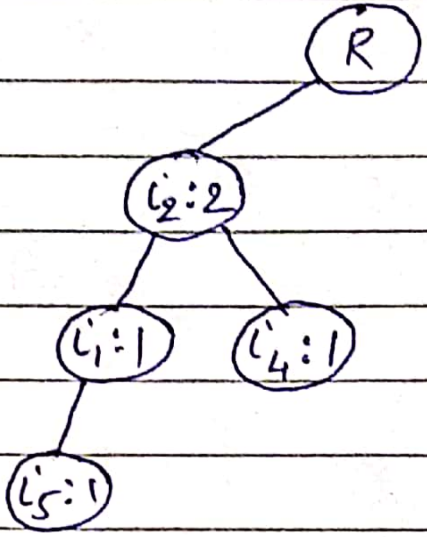
- The root of FP tree is empty. All other nodes contain an item & it's occurrence frequency with the items in predecessor nodes.

- create empty root node.

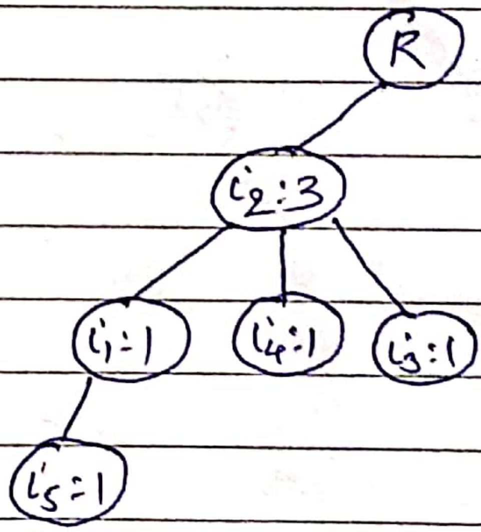
- read the 1st x'action (3rd col) & create a path in tree



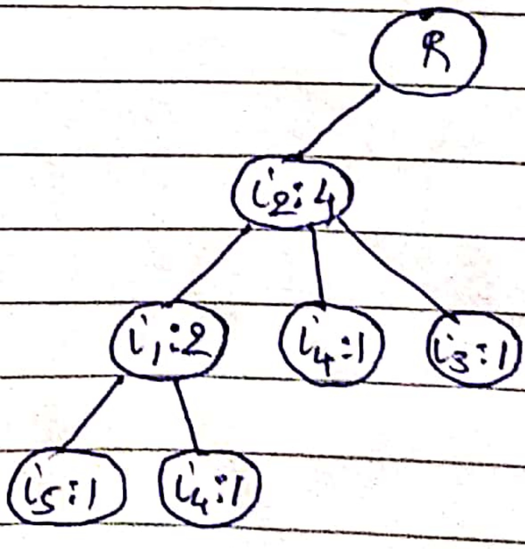
- read the 2nd x'action {i_2 i_4}. (3rd col)
as i_2 occurs under root node, increment its counter to 2, it is not followed by i_4 in the tree, so create a new path.



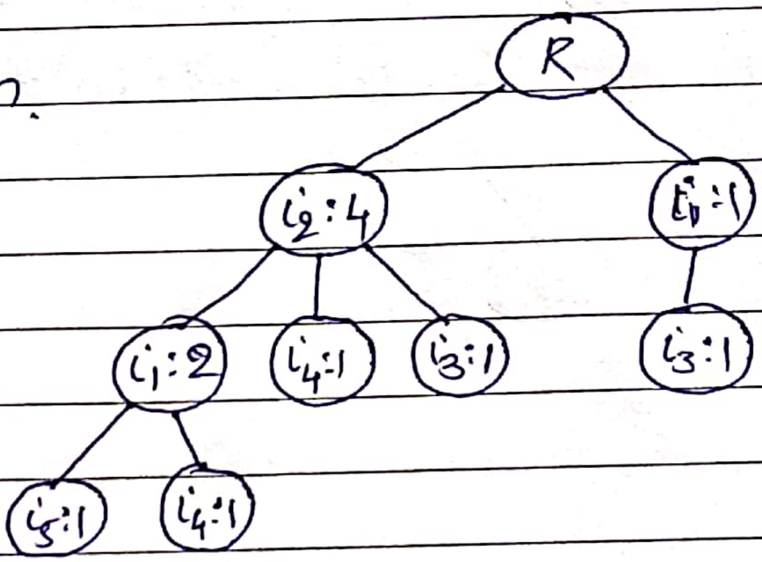
- read the 3rd x'action {i_2 i_3}.
as i_2 occurs under root node, increment its counter to 3. It is not followed by i_3, in the tree so create a new path



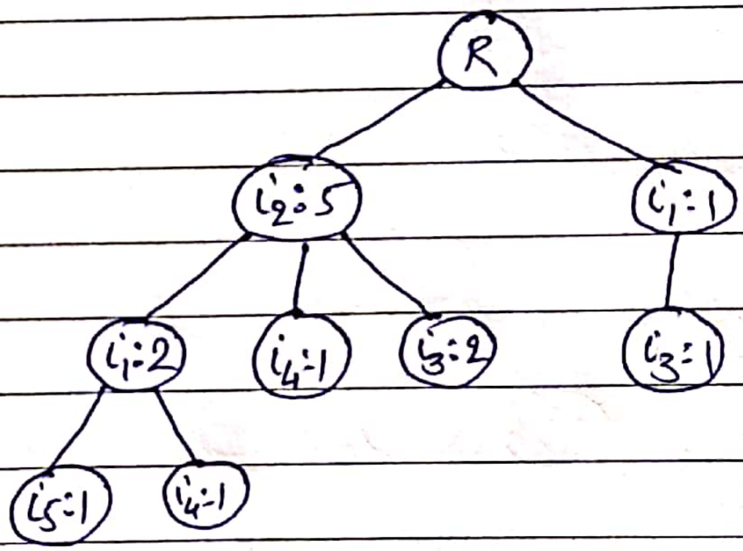
- read the 4th x'action {i_2 i_1 i_4}
i_2 occurs under root-node, increment its counter to 4.
It is followed by i_1, so increment counter to 2
It is not followed by i_4, so create a new path.



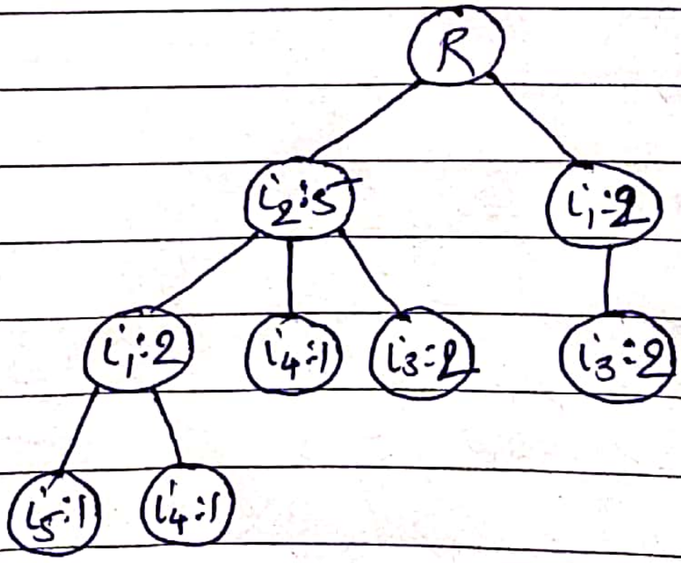
- Read the 5th x'action.
{i_1 i_3}
as i_1 doesn't occur under root node, create a new path.



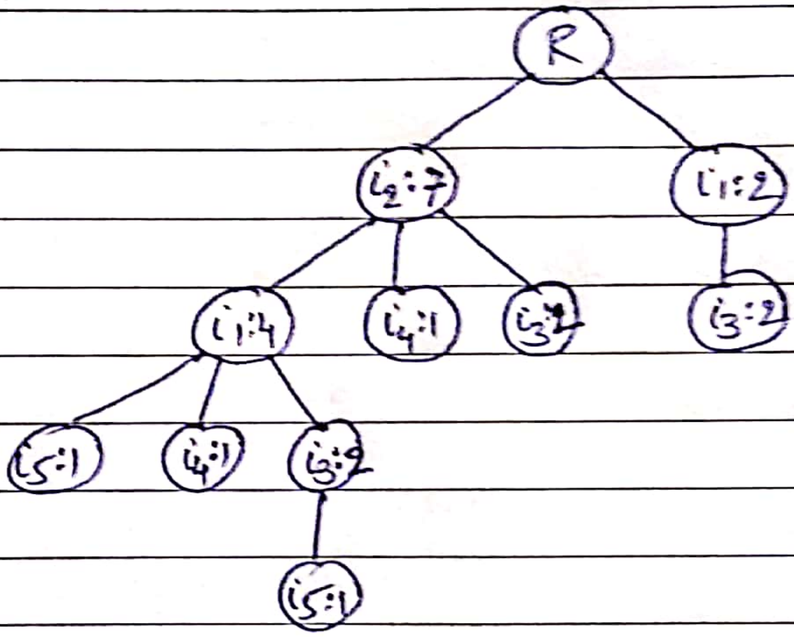
-read 6th x'action
{i_2 i_3}
as i_2 occurs under root node & it is followed by i_3 in one of the paths, increment counters in their respective nodes.



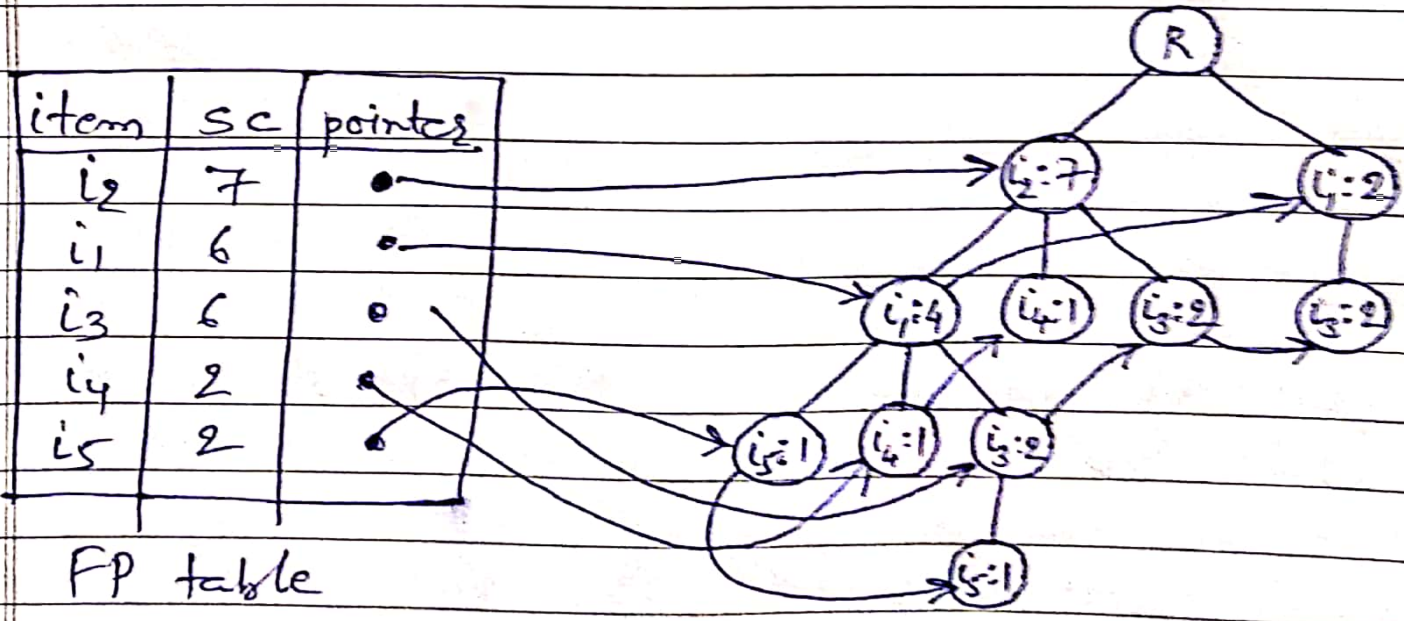
- Read the 7th x'action
{i_1 i_3}
as i_1 occurs under root node & it is followed by i_3, increment counters in their respective nodes.



- after processing x'actions T8 & T9, we get the following tree



- To process this tree, an FP table is linked to it.



item	sc	pointers
i_2	7	●
i_1	6	●
i_3	6	●
i_4	2	●
i_5	2	●

FP table

FP tree

step IV Construct conditional pattern base

- use a pointer from FP tree & write the item's occurrence with its predecessors.

$$\{i_5\} \rightarrow \{\underbrace{(i_1, i_2)}:1, (i_3, i_1, i_2):1\}$$

The item

its predecessors in FP tree

occurrence

$$\{i_4\} \rightarrow \{(i_1, i_2):1, (i_2):1\}$$

$$\{i_3\} \rightarrow \{(i_1, i_2):2, (i_2):2, (i_1):2\}$$

$$\{i_1\} \rightarrow \{(i_2):4\}$$

$$\{i_2\} \rightarrow \phi$$

step V Construct conditional FP tree

- It is basically accumulating the number of occurrences of an item with each of its predecessors in the conditional pattern base

eg $\{i_5\}$ has occurred with i_1 twice once in (i_1, i_2) & once in (i_3, i_1, i_2)

Similarly it have occurred with i_2 twice, with (i_1, i_2) also twice & with i_3 once & (i_3, i_1, i_2) once.

∴ it can be written as,

$\{i_5\} \rightarrow \{(i_1:2), (i_2:2), (i_1, i_2;2),$
$(i_3:1), (i_3, i_1:1), (i_3, i_2:1),$
$(i_3, i_1, i_2:1)\}$

Now discard the occurrences that don't satisfy the min_support (2, as given)

$\therefore \{i_5\} \rightarrow \{(i_1;2), (i_2:2), (i_1, i_2:2)\}$

Similarly for other items

$\{i_4\} \rightarrow \{(i_2:2)\}$

$\{i_3\} \rightarrow \{(i_1:4), (i_2:4), (i_1, i_2:2)\}$

$\{i_1\} \rightarrow \{(i_2:4)\}$

$\{i_2\} \rightarrow \emptyset.$

step VI find frequent patterns

- add the item with occurrences of its predecessors.

$$\therefore \{i_5\} \rightarrow \{\underbrace{(i_1, i_5:2)}_{\text{member of } L_2}, \underbrace{(i_2, i_5:2)}_{\text{member of } L_2}, \underbrace{(i_1, i_2, i_5:2)}_{\text{member of } L_3}\}$$

Similarly for other items.

$\{i_4\} \rightarrow \{(i_2, i_4 : 2)\}$

↳ L_2

$\{i_3\} \rightarrow \{(i_1, i_3 : 4), (i_2, i_3 : 4), (i_1, i_2, i_3 : 2)\}$

↳ L_2 ↳ L_2 ↳ L_3

$\{i_1\} \rightarrow \{(i_2, i_1 : 4)\}$

↳ L_2

$\therefore L_2 = \{\{i_1, i_5\}, \{i_2, i_5\}, \{i_2, i_4\}, \{i_1, i_3\},$
$\{i_2, i_3\}, \{i_2, i_1\}\}$

$L_3 = \{\{i_1, i_2, i_5\} \{i_1, i_2, i_3\}\}$

- Unlike Apriori, it is not level wise & it doesn't generate C_k.