**FP growth Algorithm:**

FP algorithm is implemented from scratch in python. Generated Condition FP trees using Bottom UP and Top Down approach:

**Example 1:**

test\_data = [['I1','I2','I5'],  
 ['I2','I3','I4'],  
 ['I3','I4'],  
 ['I1','I2','I3','I4']]  
  
**Bottom up Approach:**Condtional FPTree Root on I1 : ['Null 1', ['I2 2']]  
Condtional FPTree Root on I4 : ['Null 1', ['I3 3', ['I2 2']]]  
Condtional FPTree Root on I3 : ['Null 1', ['I2 2']]

**Top Down Approach:**Condtional FPTree Root on I1: None  
Condtional FPTree Root on I4: ['I1:1']  
Condtional FPTree Root on I3: [['I1:1'], 'I4:2']  
Condtional FPTree Root on I2: [[['I1:1'], 'I4:2'], 'I3:2', 'I1:1']

**Example 2:**

test\_data = [['I1','I2','I5'],  
 ['I2','I4'],  
 ['I2','I3'],  
 ['I1','I2','I4'],  
 ['I1','I3'],  
 ['I2','I3'],  
 ['I1','I3'],  
 ['I1','I2','I3','I5'],  
 ['I1','I2','I3']]

**Bottom up Approach:**Condtional FPTree Root on I1:['Null 1', ['I2 2']]  
Condtional FPTree Root on I4:['Null 1', ['I3 3', ['I2 2']]]  
Condtional FPTree Root on I3:['Null 1', ['I2 2']]

**Top Down Approach:**Condtional FPTree Root on I1: None  
Condtional FPTree Root on I4: ['I1:1']  
Condtional FPTree Root on I3: [['I1:1'], 'I4:2']  
Condtional FPTree Root on I2: [[['I1:1'], 'I4:2'], 'I3:2', 'I1:1']

**Example 3:**

test\_data = [['A','B'],  
 ['B','C','D'],  
 ['A','C','D','E'],  
 ['A','D','E'],  
 ['A','B','C']]  
 **Bottom up Approach:**Condtional FPTree Root on E: ['Null 1', ['A 2', ['D 2']]]  
Condtional FPTree Root on D: ['Null 1', ['C 1'], ['A 2', ['C 1']]]  
Condtional FPTree Root on C: ['Null 1', ['B 1'], ['A 2', ['B 1']]]  
Condtional FPTree Root on B: ['Null 1', ['A 2']]  
  
**Top Down Approach:**Condtional FPTree Root on E : None  
Condtional FPTree Root on D : ['E:1']  
Condtional FPTree Root on C : [['E:1'], 'D:1']  
Condtional FPTree Root on B : ['C:1']  
Condtional FPTree Root on A : [['E:1'], 'D:1', [['E:1'], 'D:1'], 'C:1', ['C:1'], 'B:2']

**Optimization and Comparison of performance:**Below is the approach followed in implementation of Fp-growth algorithm Traditional one (Bottom up approach):

* 1: Find the ordered frequent items. For items with the same frequency, order is given  
  based on the alphabetical order.
* 2: Develop the FP-tree from the above data
* 3: From FP-tree, deduce the FP-conditional tree for each item
* 4: Identify the frequent patterns

In top bottom approach: Traversed the item from top to bottom (until leaf node). For finding the Conditional FP patterns, we need to traverse the tree for all its children from top to bottom for each branch. Also, while developing FP tree, each node is linked to its another instance if they are placed in another branch. So, for traversing the children, multiple branches are traversed and selected only required items, whereas from Bottom to Top approach, where we do traverse from leaf to root, only one time traversal is fine.