

**Data Mining and Data Warehousing (CSE621)**

**Frequent Pattern Growth Algorithm**

**SUBMITTED BY:-**

AYUSH GARG

16030141CSE019

AVINASH M JAINAPUR

16030141CSE018

ABHISEKH M

16030141CSE006

Sem VI-A

**The FP-Growth Algorithm**, proposed by [Han](https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Frequent_Pattern_Mining/The_FP-Growth_Algorithm) in , is an efficient and scalable method for mining the complete set of frequent patterns by pattern fragment growth, using an extended prefix-tree structure for storing compressed and crucial information about frequent patterns named frequent-pattern tree (FP-tree). In his study, Han proved that his method outperforms other popular methods for mining frequent patterns, e.g. the Apriori Algorithm and the TreeProjection .

The FP-growth algorithm scans the dataset only twice. The basic approach to finding frequent itemsets using the FP-growth algorithm is as follows:

1 Build the FP-tree.

2 Mine frequent itemsets from the FP-tree.

The FP stands for “frequent pattern.” An FP-tree looks like other trees in computer science, but it has links connecting similar items. The linked items can be thought of as a linked list.

The FP tree is used to store the frequency of occurrence for sets of items. Sets are stored as paths

There are three basic steps to extract the frequent itemsets from the FP-tree:

1 Get conditional pattern bases from the FP-tree.

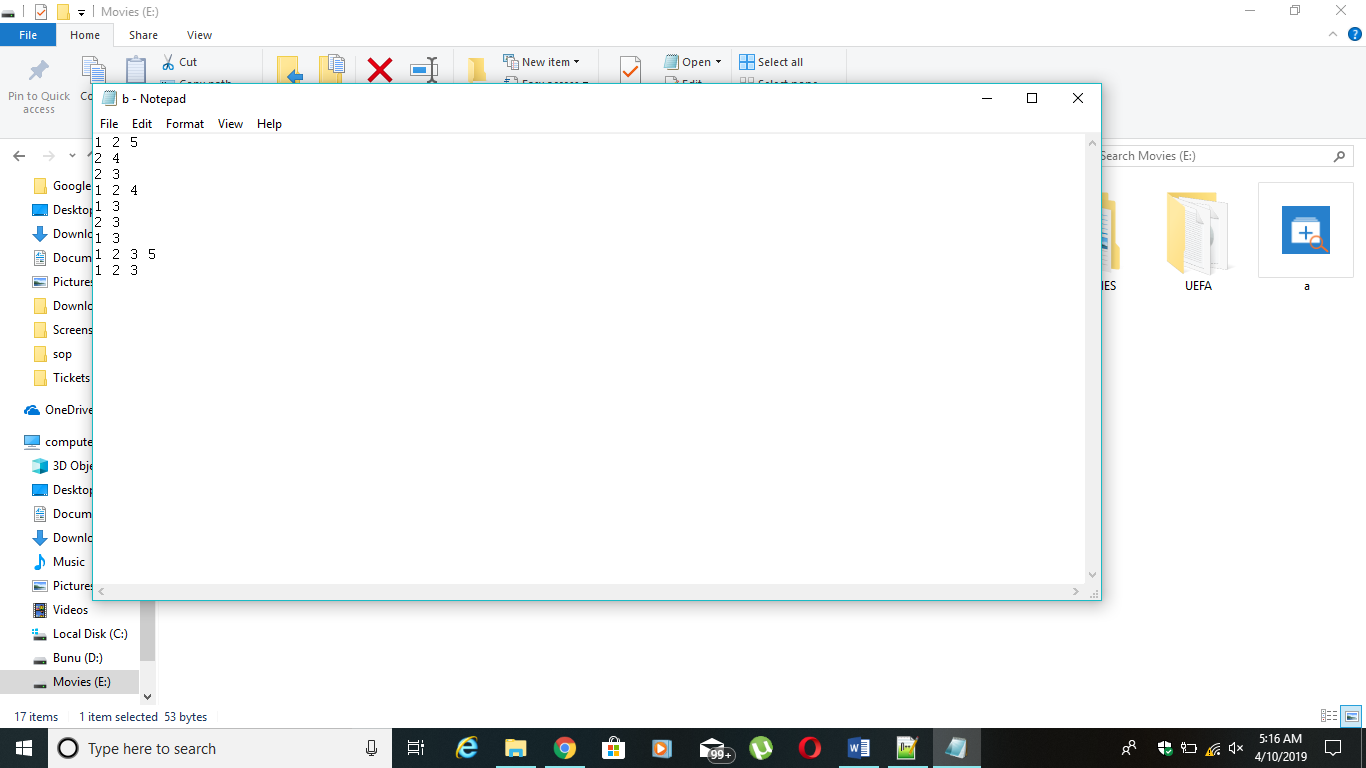
2 From the conditional pattern base, construct a conditional FP-tree.

3 Recursively repeat steps 1 and 2 on until the tree contains a single item.

The conditional pattern base is a collection of paths that end with the item you’re looking for. Each of those paths is a prefix path. In short, a prefix path is anything on the tree between the item you’re looking for and the tree root.

# Output

Dataset



A screenshot of a computer screen

Description automatically generated