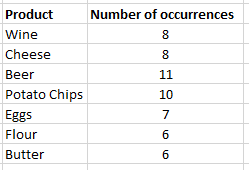
**FP Growth Alogorithm**

**Step 1 — Counting the occurrences of individual items**

The first step of the FP Growth algorithm is to count the occurrences of individual items. The below table shows the counts of each item:



The FP Growth algorithm. Counting the number of occurrences per product.

**Step 2— Filter out non-frequent items using minimum support**

You need to decide on a value for the **minimum support:***every item or item set with fewer occurrences than the minimum support will be excluded*.

In our example, let’s choose a minimum support of 7. This means that we are going to discard items Flour and Butter.

**Step 3— Order the itemsets based on individual occurrences**

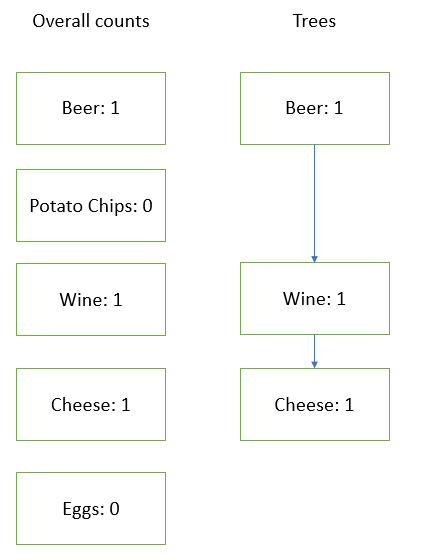
For the remaining items, we will create an ordered table. This table will contain the items that have not been rejected yet, and the items inside a transaction will be ordered based on individual product occurrence.



The FP Growth Algorithm. Ordering itemsets based on individual product occurrence.

**Step 4— Create the tree and add the transactions one by one**

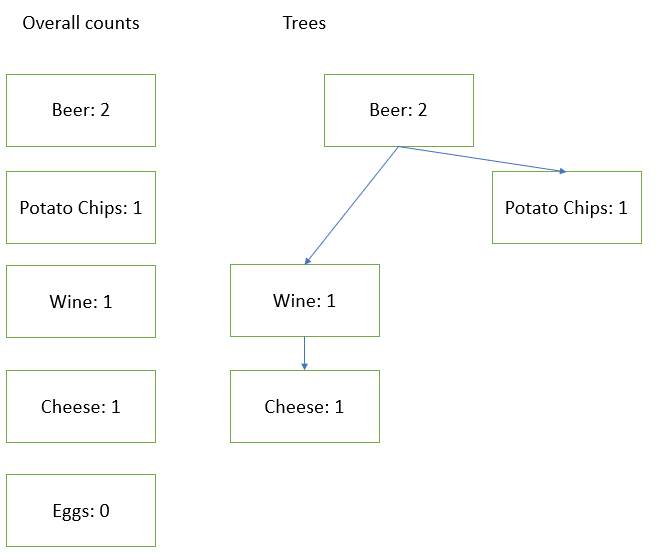
Now, we can create the tree starting with the first transaction. Each product is a node in the tree, as follows:



The FP Growth Algorithm. The tree after adding only the first transaction.

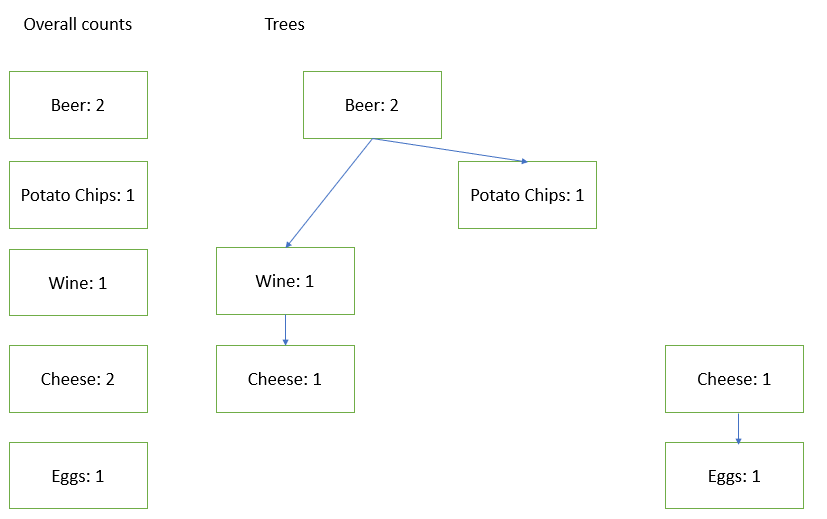
Note that we also add a count to each product, which we will use for counting later on.

We can now move on to the second transaction:



The FP Growth Algorithm. The tree after adding the first two transactions.

Let’s now see what happens when we add the third transaction:



The FP Growth algorithm. The tree after adding the third transaction.

The third transaction did not contain Beer, unlike the two first transactions. Therefore, it was not possible to link it directly to the first tree. It has a separate starting node that you can get to via the main cheese node.

Once this FP tree is constructed, it is much faster to traverse it and find information on the most frequent itemset.