National University of Computer & Emerging Sciences (NUCES),
Islamabad Department of Computer Science

CS-218 Data Structures (Spring 2020)
Assignment #01

Finding the Frequent itemset in Grocery Store

Assignment Description:

This assignment will give you basic insight into using Apriori algorithm. Apriori is use for finding the frequent item set in transaction. For example, in Supermarket store where customers can buy different categories of items. There is always a pattern for what a customer buy. This pattern changes according to the buyer. For example, if a buyer is a Player, and he buys products such as Bat, Ball then, its most probable that he will purchase a tape too. But if the buyer is a mother having babies, she will buy baby products such as milk and diapers. In short, every buyer's transaction involves a pattern. Our goal is to find those patterns in these transactions. Profit is automatically generated if the relationship is found between the items purchased in different transactions.
Apriori algorithm was the first algorithm that was proposed for frequent itemset mining. Let's understand Apriori algorithm using an example step by step:

You are given the grocery store dataset. Text file contains:

1. The first row of the file contains *Support threshold* e.g. 0.5.
2. The second row contains total number of transactions.
3. Followed by transactions. Each transaction contains different set of items separated by comma. E.g. Bread, Cheese, Egg, Juice etc

Example:

0.5
6
Bread,Cheese,Egg,Juice
Bread,Chesse,Juice
Bread,Milk,Yougrt
Bread,Juice,Milk
Cheese,Juice,Milk
Bread,Egg,Cheese,Juice

Support threshold is 0.5 and there are total no of 6 transaction and the first transaction is **Bread,Cheese,Egg,Juice** in the above example.

Support threshold determines, is the item popular or not.

Step 01:
Read the file and store all the transactions in 2D char Array. And evaluate *Support threshold=0.5*No of transaction*. So, you get threshold 0.5 * 6 = 3.
Note: File should be read once.

Step 02:
Now you have to find the frequency of all the items using 2D char Array.

Items	Frequency
Bread	5
Cheese	4
Egg	2
Juice	5
Milk	3
Yogurt	1

Table 01: Frequency of each item

Step 03:
The set of 1st – itemset whose occurrence is satisfying the **Support threshold** are determined. Only those items which count more than or equal to **Support threshold** are taken ahead for the next iteration and the others are pruned (deleted). E.g. Egg and Yogurt have frequencies less than 3. Remove Items that does not meet the minimum support threshold and sort them. The output should look like this.

Items	Frequency
Bread	5
Juice	5
Cheese	4
Milk	3

Table 02: 1st - ItemSet

Step 04:
Next, 2-item pair candidate frequencies are discovered. The 2-item pairs are generated by forming a group of 2 items using **1st – ItemSet** (Table 02**)** and **2D char Array**.

Items	Frequency
Bread, Cheese	3
Bread, Juice	4
Bread,Milk	2
Cheese,Juice	4
Cheese,Milk	1
Juice,Milk	2

Table 03: Frequency of 2-item pairs

Step 05:
The 2-itemset candidates are pruned (deleted) using **Support threshold** value and then sort them. Now the table will have 2 –item sets with **Support threshold**.

Item	Frequency
Bread,Juice	4
Bread, Cheese	3
Cheese,Juice	3

Table 04: 2nd – ItemSet

Step 06:

The 3-item pair candidates with frequencies are generated by grouping pair of 3 items using **1st – ItemSet (**Table 02**) and 2D char Array**.

Item	Frequency
Bread, Cheese,Juice	3
Bread, Cheese,Milk	0
Bread, Juice,Milk	1
Cheese,Juice,Milk	1

Table 05: Frequency of 3-item pairs

Step 07:

The 3-itemset candidates are pruned (deleted) using **Support threshold** value.

Item	Frequency
Bread, Cheese,Juice	3

Table 06: 3nd – ItemSet

You are only required to generate 1st , 2nd & 3rd ItemSets.

Output files & Marks distribution:

1. **Generate 1st – ItemSet (**as showed in Table 02**)** and write on 1-ItemSet.txt (**Marks 60**)
2. **Generate 2nd – ItemSet (**as showed in Table 04**)** and write on 2-ItemSet.txt (**Marks 20**)
3. **Generate 3rd – ItemSet (**as showed in Table 06**)** and write on 3-ItemSet.txt (**Marks 20**)

Note: Path of the file must be same as the Project path.

Submission Criteria & Guidelines:

1. **Submission:** You are required to use Visual Studio 17 or above for the assignment. Combine all your work (only .cpp , .h) in one zip file. Main.cpp must be submitted. DO NOT SUBMIT COMPLETE PROJECT. Name the .zip file as ROLL_NUM_A_01.zip (e.g. **18i-0001_A_01.zip**). Submit zip file on slate or in classroom within given deadline. Failure to submit according to above format would result in **ZERO** marks.
2. Path of files must be same as the Project path. DO NOT USE ABSOLUTE PATH. If the files are not generated in required path no marks will be awarded.
3. Do not change the name of the dataset (GroceryStore.txt). Path of files must be same as the Project path.
4. Use Classes. If the assignment is **classes less** no marks will be awarded.

5. Use Class templates for storing ItemSets and sorting. If Class templates are not used, 30 marks will be deducted.
6. You are only required to generate 1st , 2nd & 3rd ItemSets.
7. File should be read once.
8. Do not use STLs, String library or any built-in functions except char Array functions (e.g. strcmp , strlen , …)
9. If the required output is generated, you will be awarded full marks. Failing to generate the correct output will result in zero marks(black box checking only).
10. If we unable to compile because of **syntax error** no marks will awarded.
11. Plagiarism cases will be dealt strictly. If found plagiarized, both the involved parties will be awarded zero marks in this assignment. **Copying from the internet is the easiest way to get caught!**
12. **Deadline:** Deadline to submit assignment is **17th February, 2020 10:00 AM**. Late submission with 30 marks deduction will be accepted till **18th February, 2020 10:00 AM**. No submission will be considered for grading after **18th February, 2020 10:00 AM**. Correct and timely submission of assignment is responsibility of every student; hence no relaxation will be given to anyone.