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

CS-2001 Data Structures (Fall 2021)
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 e.g., 6
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,Cheese,Juice
Bread,Milk-,Yougrt
Bread,+Juice@,Milk
Cheese,Juice,Milk1
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.
“A set of items that appears in many baskets is said to be “frequent.” To be formal, we assume there is a number **S**, called the support threshold. If **I** is a set of items, the support for **I** is the number of baskets(transactions) for which I is a subset. We say I is frequent if its support is S or more” (reference: Knowledge Discovery and Data Mining for Predictive Analytics Book). In our example S= 0.5 and 6 is number of transactions. So, item is frequent if it repeats in 0.5 * 6 = 3 or more transactions.

Step 01:
Read the file and store all the transactions in Link Lists (TransactionLL). Each transaction (e.g. **Bread,Cheese,Egg,Juice**) will be list of items. And transactions are itself list.

Note: Since File I/O is a very costly operation, you are required to traverse the file once (multiple traversals of file result in **ZERO** marks).

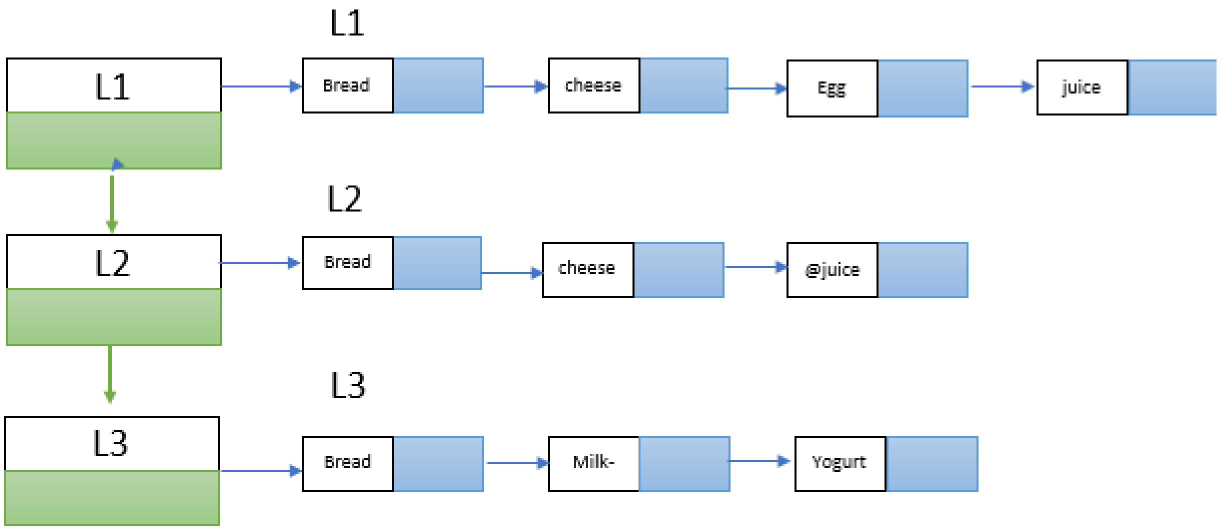


Figure 01: TransactionLL

Step 02:
Perform preprocessing on TransactionLL:

a. Remove all the punctuations marks and numbers (.~ ! @ # $ % ^ &amp; * ( ) _ + = “ ; : / ? extra spaces or 0 to 9 digits.
   Note: Remember
   1) Output must only contain alphabets and space characters.
   2) There must be single space between two words (if the item name is composed of muti word e.g., black pepper).

b. Convert all upper-case letters to lower case (because “Bread” and “bread” both are same item).

Step 03:
you have to find the frequency of all the items using TransactionLL (updated in Step 02).

Items	**Frequency**
bread	**5**
cheese	**4**
egg	**2**
juice	**5**
milk	**3**
yogurt	**1**

Table 01: Frequency of each item

Step 03:
Now evaluate *Support threshold=0.5*No of transaction*. So, you get threshold 0.5 * 6 = 3. 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 (see Table 01). Remove Items that does not meet the minimum support threshold and sort them. **Also remove those items from TransactionLL** (Because we do not need non frequent items for next iterations).

The output should look like this.

Items	**Frequency**
bread	**5**
juice	**5**
cheese	**4**
milk	**3**

Table 02: 1st - ItemSet

bread,cheese,juice
bread,cheese,juice
bread,milk
bread,juice,milk
cheese,juice,milk
bread,cheese,juice

Table 03: TransactionLL

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 generate frequency using updated TransactionLL (Table 03).

Items	**Frequency**
bread,juice	**4**
bread,cheese	**3**
bread,milk	**2**
juice,cheese	**4**
juice,milk	**2**
cheese,milk	**1**

Table 03: Frequency of 2-item pairs

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**
juice,cheese	**4**
bread,cheese	**3**

Table 04: 2nd – ItemSet

Step 05:
The 3-item pair candidates with frequencies are generated by grouping pair of 3 items using **1st – ItemSet (**Table 02**)** and TransactionLL (Table 03).

Item	**Frequency**
bread,juice,cheese	**3**
bread,juice,milk	**1**
bread,cheese,milk	**0**
juice,cheese,milk	**1**

Table 05: Frequency of 3-item pairs

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

Item	**Frequency**
bread,juice,cheese	**3**

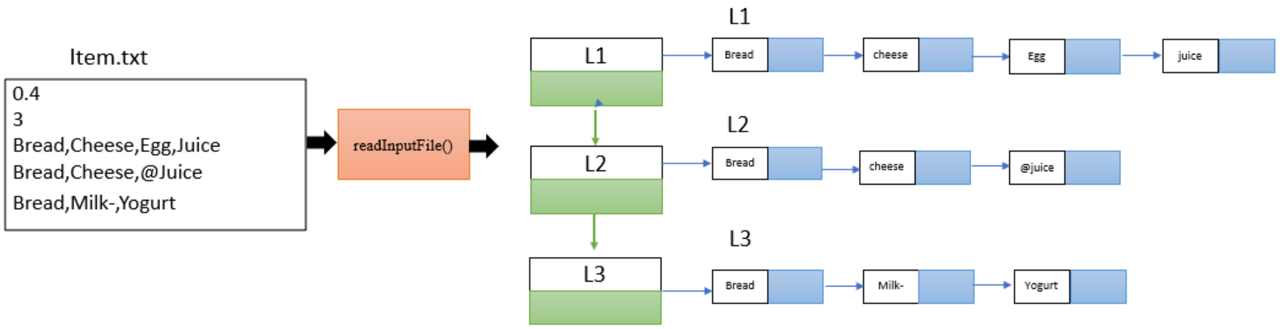
Table 06: 3nd – ItemSet

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

By carefully analyzing these Itemsets, we can come up with a more detailed explanation of how to make business decisions in retail environments. Now, we know Bread, Cheese, Juice and Milk are frequent items and would place “Bread”, “Cheese” and “Juice” beside each other.

Function Prototypes:

1. void readInputFile (char* inputFilePath)

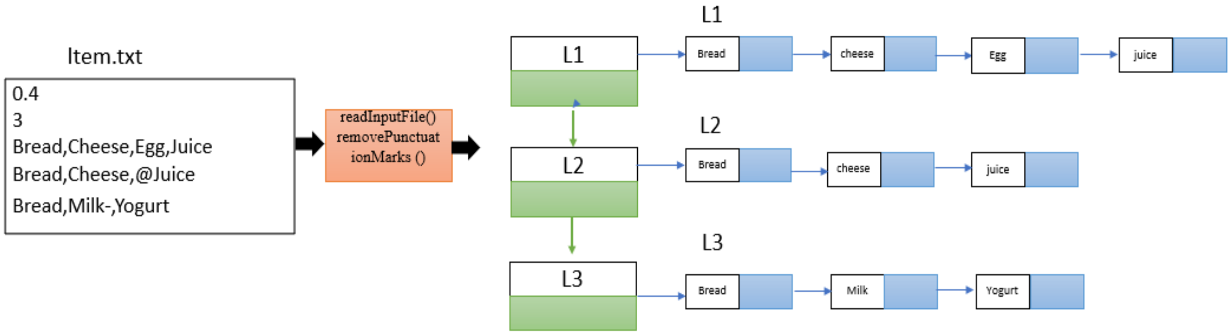


Item.txt

```
0.4
3
Bread,Cheese,Egg,Juice
Bread,Cheese,@Juice
Bread,Milk-,Yogurt
```

Des: This function will be used to load the data from input file onto the TransactionLL (Data Structure).

2. void removePunctuationMarks ()

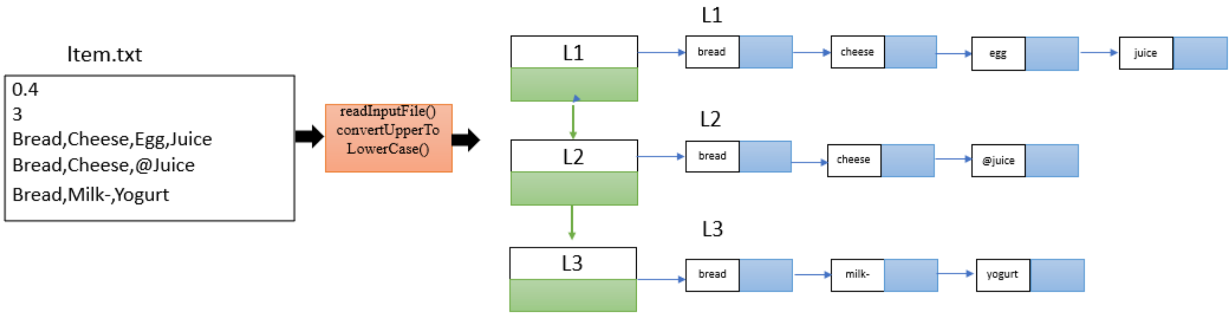


Item.txt

```
0.4
3
Bread,Cheese,Egg,Juice
Bread,Cheese,@Juice
Bread,Milk-,Yogurt
```

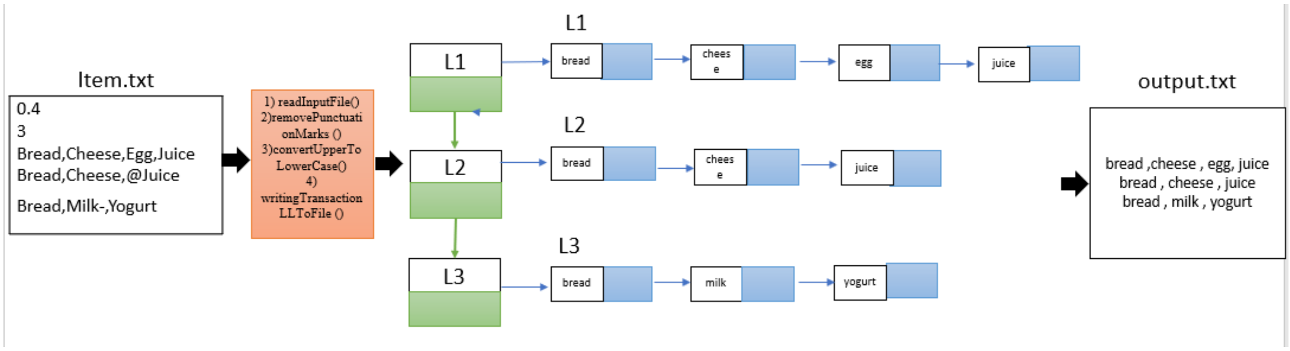
Des: This function will remove punctuation marks and numbers according to Step 02.a from a double Linklist

3. void convertUpperToLowerCase ()



Des: This function will convert upper case letters to lower case letters according to Step 02.b from a double Linklist

4. void writingTransactionLLToFile (char* outputFilePath)



Des: This function will write all the transaction (current TransactionLL) onto the file (Path: char* outputFilePath). Items are separated by comma and transitions in different lines.

5. void generateFirstItemSet(char* LL_frequency)

Des: Refer to Step 03 and write 1st – itemset onto the file (Path: char* LL_frequency).

LL_frequency = Items and their frequency (only those items whose frequency is greater than threshold

Note: The item must be in sorted order based on their frequencies (Higher to lower))

6. void generateSecondItemSet(char* frequency_outputfile)

Des: Refer to Step 04 and write 2nd – itemset onto the file (Path: char* frequency_outputfile).

7. void generateThirdItemSet(char* frequency_outputfile)

Des: Refer to Step 05 and write 3rd – itemset onto the file (Path: char* frequency_outputfile).

Note:

1) USE TEMPLATES to implement all the functions **and structure**
2) Do not use arrays

Submission Criteria & Guidelines:

1. **Submission:** You are required to use Visual Studio 19 or above for the assignment. Combine you all work in one .h file named ROLL_NUM_A_01.h (e.g., **20i-0001_A_01.zip**). DO NOT SUBMIT COMPLETE PROJECT. Move .h file to folder as ROLL_NUM_A_01 then .zip file. Submit zip file 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. Code must be generic.
5. Use Structure/Class templates. If the class templates did not use no marks will be awarded.
6. Wherever array can be used, use Link List. Array is not allowed.
7. You are only required to generate 1st, 2nd & 3rd ItemSets.
8. File should be read once.
9. Do not use STLs, String library or any built-in functions.
10. 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).
11. If we unable to compile because of **syntax error** no marks will awarded.
12. 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!**
13. **Deadline:** Deadline to submit assignment is **17 October 2021**. Correct and timely submission of assignment is responsibility of every student; hence no relaxation will be given to anyone.