Homework 1
COL 761

* You can do the homework in groups of 3

* Due: **August 23, 2018 11:59PM**.

* **30% penalty** for each late day

---

1. This question is on frequent itemset mining. Implement **Apriori Algorithm** and **FP-tree** to mine frequent itemsets. Apply it on the Dataset: http://fimi.ua.ac.be/data/retail.dat. Details about the dataset can be found at http://fimi.ua.ac.be/data/retail.pdf. You may assume all items are integers.

A. Please name your file `<RollNo>.sh`. For example, if 2014CS50288 is your roll number, your file should be named `2014CS50288.sh`. Executing the command: `sh <RollNo>.sh <inputfile> X -apriori <filename>` should generate a file `<filename>.txt` containing the frequent itemsets at `>=X%` support threshold with the Apriori algorithm. Similarly, executing the command: `sh RollNo.sh <inputfile> X -fptree <filename>` should generate a file `<filename>.txt` containing the frequent itemsets using FP-tree algorithm. Notice that `X` is in percentage and not the absolute count. Your implementations must ensure that the transactions are not loaded into main memory. However, the frequent patterns and candidate sets can be stored in memory. **(20+20=40 points)**

`<filename>.txt` should strictly follow the following format:

1.Each frequent itemset must be on a new line.

2.The items in each itemset must be space separated.

Your grade will be **(F-score_apriori)*20 + (F-score_fptree)*20**.

B. Compare the performance of Apriori algorithm with FP-tree. The FP-tree implementation and Apriori implementation must be in the same language. Executing the command: `sh <RollNo>.sh <inputfile> -plot` should generate a plot using `matplotlib` where the x-axis varies the support threshold and y-axis shows the corresponding running times. It should plot the running times at support thresholds of 1%, 5%, 10%, 25%, 50%, and 90%. Plot at these support values and explain the results that you observe. **(10 points for correct plot + 10 points for explanation=20 points)**

C. Efficiency Competition: We will have a competition among all submitted implementations of the FP-tree. The fastest would get full points. If your algorithm is X% slower than the fastest, then you would get X% of full points. You would be in this competition only if you get full points in part (A), i.e., you have the correct implementation of the FP-tree algorithm. **(20 points)**

Files you need to submit:

1. A Readme file explaining all files you have bundled. In addition, it should also contain explanation for part B.
2. A compilation script `compile.sh`, which we will use to compile all your submitted source code.
3. `<Rollno>.sh` is the main script that will be used in parts A, B, and C.
4. Relevant files for plot generation

Anti-Plagiarism Policy:

Do not copy code from any other or past submissions or from the Internet. We have downloaded all available libraries of Apriori algorithm and FP-tree from the web and all submitted codes will be checked against these as well as those submitted in this homework. Any plagiarised code will result in an F grade for the course straight away.