Homework #2: Finding Frequent Itemsets

Due: February 24, Friday
100 points

In this homework, you are asked to implement SON algorithm "son.py" in **Apache Spark** using Python. Recall that given a set of baskets, SON algorithm divides them into chunks/partitions and then proceed in two stages. First, local frequent itemsets are collected, which form candidates; next, it makes second pass through data to determine which candidates are globally frequent.

In this homework, please implement an Apriori algorithm for stage one. You may find Python itertools package to be useful in your implementation:
https://docs.python.org/2/library/itertools.html#itertools.combinations

Requirements: You must use mapPartitions() that invokes Apriori to find frequent itemsets in each partition.

Format of execution:

```
bin/spark-submit <FirstName>_<LastName>_ son.py baskets.txt .3 output.txt
```

baskets.txt is a text file which contains a basket (a list of comma-separated item numbers) per line. For example

```
1,2,3
1,2,5
1,3,4
2,3,4
1,2,3,4
2,3,5
1,2,4
1,2
1,2,3
1,2,3,4,5
```

.3 is a minimum support ratio (that is, for an itemset to be frequent, it should appear in at least 30% of the baskets).

output.txt is the name of output file.

Format of output:

You should save all frequent itemsets into one text file. Each line of the file contains one itemset (a list of comma-separated item numbers). The order doesn't matter. For example,

```
4
1,3,4
1,2,3
2
1,3
2,4
2,3
1
2,3,4
1,4
3
3,4
1,2,4
2,5
1,2
5
```

Submissions:

Name your python script as <FirstName>_<LastName>_ son.py and submit to blackboard by the due time.

Notes:

1. The program takes 3 arguments:
    - Input file
    - Minimum support ratio
    - Output file
2. All basket item are positive integers.
3. Make sure to follow the output format (The order of output itemset doesn't matter) and the submission naming format. If you don't follow either one or both of them, 20% points will be deducted.
4. A standalone Apriori need to be implemented in stage one. The second stage needs to be done in Spark (using mapPartitions()) too.
5. We will be using Moss for plagiarism detection.