MIDS UC Berkeley - Machine Learning at Scale
DATSCIW261 ASSIGNMENT #3

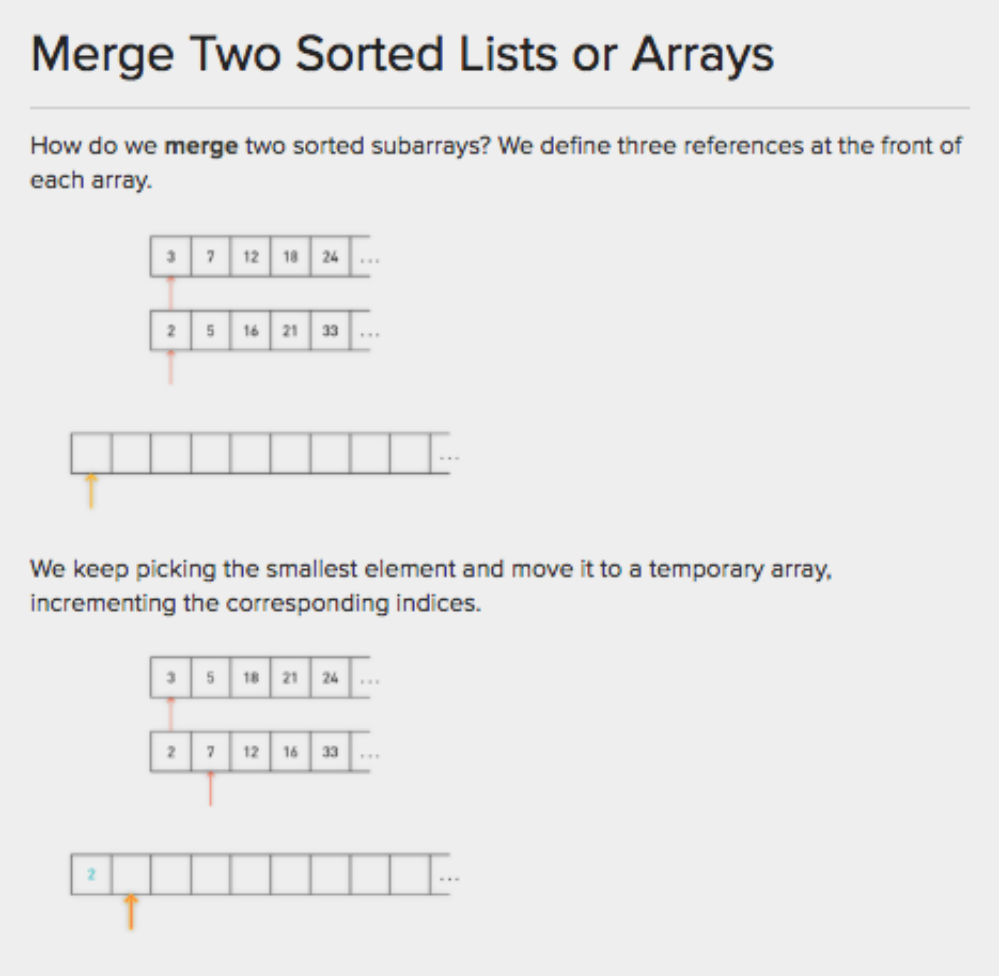
[James Gray](https://github.com/jamesgray007)
jamesgray@ischool.berkeley.edu
Time of Initial Submission: 12:45 PM US Central, Sunday, June 5, 2016
Time of **Resubmission**:
W261-1, Spring 2016
Week 3 Homework

References for this Assignment

- x

HW3.0

- ***How do you merge two sorted lists/arrays of records of the form [key, value]? Where is this used in Hadoop MapReduce? [Hint within the shuffle]***



The "merge sort" algorithm merges two sorted lists into a third list. Pointers are set at the start of each list. The pointer picks off the first number of the first list and another pointer picks off the first number of the second list. The smaller value is copied to the third list and the pointer is moved to the right of the list with the smaller number and the pointer of the third list (merged list) is also moved. This process is repeated until all of the two lists have been read.

- ***What is a combiner function in the context of Hadoop?***

A combiner is a mapper-side function that consolidates key-value pairs with the same key before it enters the Hadoop shuffle process. This reduces network congestion between mappers and reducers.

- ***Give an example where it can be used and justify why it should be used in the context of this problem.***

For example, if we were computing a word count on a document corpus instead of emitting key-value pairs as "word, 1", we could "reduce" on the mapper side by counting the frequency of each word before sending the data to the reducer. This would reduce network traffic.

- ***What is the Hadoop shuffle?***

The Hadoop shuffle is the heart of the map-reduce framework that transfers and synchronizes data between the map and reduce process. There are three parts to the shuffle:

1. partition - the partitioner organizes the mapper output by key and creates a file for each reducer.
2. sort - after the data is partitioned into files the sort function sorts the data.
3. combine - the combine function uses the sorted data and combines key-value pairs with the same key.

The data is directed to the reducer(s) once the three shuffle steps have been completed. See Wikipedia (https://en.wikipedia.org/wiki/Merge_sort) for more information.

HW3.1 - Consumer Complaints dataset - Use Counters to do EDA

Counters are lightweight objects in Hadoop that allow you to keep track of system progress in both the map and reduce stages of processing. By default, Hadoop defines a number of standard counters in "groups"; these show up in the jobtracker webapp, giving you information such as "Map input records", "Map output records", etc.

While processing information/data using MapReduce job, it is a challenge to monitor the progress of parallel threads running across nodes of distributed clusters. Moreover, it is also complicated to distinguish between the data that has been processed and the data which is

yet to be processed. The MapReduce Framework offers a provision of user-defined Counters, which can be effectively utilized to monitor the progress of data across nodes of distributed clusters.

Use the Consumer Complaints Dataset provide here to complete this question:

```
https://www.dropbox.com/s/vbalm3yva2rr86m/Consumer_Complaints.csv?dl=0
```

The consumer complaints dataset consists of diverse consumer complaints, which have been reported across the United States regarding various types of loans. The dataset consists of records of the form:

Complaint ID,Product,Sub-product,Issue,Sub-issue,State,ZIP code,Submitted via,Date received,Date sent to company,Company,Company response,Timely response?,Consumer disputed?

User-defined Counters

Now, let's use Hadoop Counters to identify the number of complaints pertaining to debt collection, mortgage and other categories (all other categories get lumped into this one) in the consumer complaints dataset. **Basically produce the distribution of the Product column in this dataset using counters (limited to 3 counters here).**

Hadoop offers Job Tracker, an UI tool to determine the status and statistics of all jobs. Using the job tracker UI, developers can view the Counters that have been created. **Screenshot your job tracker UI as your job completes and include it here. Make sure that your user defined counters are visible.**

HW3.1 - Mapper

In [3]:
```
%%writefile mapper31.py
#!/usr/bin/python
## mapper31.py
## Author: James Gray
## Description: mapper code for HW3.1

import sys
for line in sys.stdin:
    line=line.strip()
    product=line.split(',')[1] #extract product field from second fiel

    # populate 3 counters based on Consumer_Complaints.csv product typ
    if product=='Debt collection':
        sys.stderr.write("reporter:counter:Debt,Total,1\n")
    if product=='Mortgage':
        sys.stderr.write("reporter:counter:Mortgage,Total,1\n")
    else:
        sys.stderr.write("reporter:counter:Other,Total,1\n")
    print product+'\t1'
```

```
Overwriting mapper31.py
```

HW3.1 - Reducer

In [4]:
```
%%writefile reducer31.py
#!/usr/bin/python
## reducer31.py
## Author: James Gray
## Description: reducer code for HW3.1

import sys
for line in sys.stdin:
    line=line.strip()
    print line
```

```
Writing reducer31.py
```

In [5]:
```
# set file priveleges to execute script
!chmod a+x reducer31.py
!chmod a+x mapper31.py
```

HW3.1 - Populate HDFS

In [8]:
```
# transfer Consumer_Complaints.csv into HDFS

#!hdfs dfs -put Consumer_Complaints.csv /user/graymatter/consumer_compl

!hdfs dfs -ls /user/graymatter/*
```

```
16/05/26 14:29:35 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
-rw-r--r--   1 jamesgray supergroup   50906486 2016-05-26 14:29 /use
r/graymatter/consumer_complaints.csv
-rw-r--r--   1 jamesgray supergroup     202254 2016-05-23 18:10 /use
r/graymatter/enronemail_1h.txt
Found 2 items
-rw-r--r--   1 jamesgray supergroup          0 2016-05-24 23:07 /use
r/graymatter/hw21-output/_SUCCESS
-rw-r--r--   1 jamesgray supergroup     108879 2016-05-24 23:07 /use
r/graymatter/hw21-output/part-00000
Found 2 items
-rw-r--r--   1 jamesgray supergroup          0 2016-05-21 14:46 /use
r/graymatter/hw21-output.txt/_SUCCESS
-rw-r--r--   1 jamesgray supergroup     108879 2016-05-21 14:46 /use
r/graymatter/hw21-output.txt/part-00000
Found 2 items
-rw-r--r--   1 jamesgray supergroup          0 2016-05-21 17:22 /use
r/graymatter/hw22-output/_SUCCESS
-rw-r--r--   1 jamesgray supergroup         14 2016-05-21 17:22 /use
r/graymatter/hw22-output/part-00000
Found 2 items
-rw-r--r--   1 jamesgray supergroup          0 2016-05-21 17:27 /use
r/graymatter/hw221-output/_SUCCESS
-rw-r--r--   1 jamesgray supergroup     226846 2016-05-21 17:27 /use
r/graymatter/hw221-output/part-00000
Found 2 items
-rw-r--r--   1 jamesgray supergroup          0 2016-05-24 23:37 /use
r/graymatter/hw221_job1-output/_SUCCESS
-rw-r--r--   1 jamesgray supergroup      53528 2016-05-24 23:37 /use
r/graymatter/hw221_job1-output/part-00000
Found 2 items
-rw-r--r--   1 jamesgray supergroup          0 2016-05-24 23:52 /use
r/graymatter/hw221_job2-output/_SUCCESS
-rw-r--r--   1 jamesgray supergroup      53528 2016-05-24 23:52 /use
r/graymatter/hw221_job2-output/part-00000
Found 2 items
-rw-r--r--   1 jamesgray supergroup          0 2016-05-24 20:41 /use
r/graymatter/hw23_job1-output/_SUCCESS
-rw-r--r--   1 jamesgray supergroup     177210 2016-05-24 20:41 /use
r/graymatter/hw23_job1-output/part-00000
Found 2 items
-rw-r--r--   1 jamesgray supergroup          0 2016-05-24 20:41 /use
r/graymatter/hw23_job2-output/_SUCCESS
-rw-r--r--   1 jamesgray supergroup         58 2016-05-24 20:41 /use
r/graymatter/hw23_job2-output/part-00000
Found 2 items
-rw-r--r--   1 jamesgray supergroup          0 2016-05-25 12:07 /use
r/graymatter/hw24_job1-output/_SUCCESS
-rw-r--r--   1 jamesgray supergroup     239095 2016-05-25 12:07 /use
r/graymatter/hw24_job1-output/part-00000
Found 2 items
-rw-r--r--   1 jamesgray supergroup          0 2016-05-25 12:07 /use
```

```
r/graymatter/hw24_job2-output/_SUCCESS
-rw-r--r--   1 jamesgray supergroup         52 2016-05-25 12:07 /use
r/graymatter/hw24_job2-output/part-00000
Found 2 items
-rw-r--r--   1 jamesgray supergroup          0 2016-05-25 12:55 /use
r/graymatter/hw25_job1-output/_SUCCESS
-rw-r--r--   1 jamesgray supergroup      52270 2016-05-25 12:55 /use
r/graymatter/hw25_job1-output/part-00000
-rw-r--r--   1 jamesgray supergroup      98879 2016-05-20 21:04 /use
r/graymatter/integers.txt
```

HW3.1 - Execute MR Job

In [12]:
```
# delete output directory
!hdsf dfs -rm -r /user/graymatter/hw31-output

!hadoop jar /usr/local/Cellar/hadoop/2.7.1/libexec/share/hadoop/tools/l
-files mapper31.py,reducer31.py -mapper mapper31.py -reducer reducer31.
 input /user/graymatter/consumer_complaints.csv -output /user/graymatte
```

```
16/05/26 14:39:45 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
16/05/26 14:39:45 INFO Configuration.deprecation: session.id is depr
ecated. Instead, use dfs.metrics.session-id
16/05/26 14:39:45 INFO jvm.JvmMetrics: Initializing JVM Metrics with
processName=JobTracker, sessionId=
16/05/26 14:39:45 INFO jvm.JvmMetrics: Cannot initialize JVM Metrics
with processName=JobTracker, sessionId= - already initialized
16/05/26 14:39:46 ERROR streaming.StreamJob: Error Launching job : O
utput directory hdfs://localhost:9000/user/graymatter/hw31-output al
ready exists
Streaming Command Failed!
```

HW3.1 - Counter Output

```
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=9
                Total committed heap usage (bytes)=74
        Debt
                Total=44372
        Mortgage
                Total=125752
        Other
                Total=187161
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=50906486
```

HW3.2 - Analyze the performance of your Mappers, Combiners and Reducers using Counters

For this brief study the Input file will be one record (the next line only): **foo foo quux labs foo bar quux**

Part 1

Perform a word count analysis of this single record dataset using a Mapper and Reducer based WordCount (i.e., no combiners are used here) using user defined Counters to **count up how many time the mapper and reducer are called**. What is the value of your user defined Mapper Counter, and Reducer Counter after completing this word count job. The answer should be 1 and 4 respectively. Please explain. Please use mulitple mappers and reducers for these jobs (at least 2 mappers and 2 reducers).

Part 2

Perform a word count analysis of the Issue column of the Consumer Complaints Dataset using a Mapper and Reducer based WordCount (i.e., no combiners used anywhere) using user defined Counters to count up how many time the mapper and reducer are called. **What is the value of your user defined Mapper Counter, and Reducer Counter after completing your word count job**.

Part 3

Perform a word count analysis of the Issue column of the Consumer Complaints Dataset using a Mapper, Reducer, and standalone combiner (i.e., not an in-memory combiner) based WordCount using user defined Counters to **count up how many time the mapper, combiner, reducer are called. What is the value of your user defined Mapper Counter, and Reducer Counter after completing your word count job.**

Part 4

Using a single reducer:

- What are the top 50 most frequent terms in your word count analysis? Present the top 50 terms and their frequency and their relative frequency.
- If there are ties please sort the tokens in alphanumeric/string order. Present bottom 10 tokens (least frequent items). Please use a combiner.

Populate HDFS with file

In [24]:
```
# the text "foo foo quux labs foo bar quux" was added to "singleline.tx

#!hdfs dfs -put singleline.txt /user/graymatter/singleline.txt

!hdfs dfs -cat /user/graymatter/singleline.txt
```

```
16/05/27 07:58:04 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
foo foo quux labs foo bar quux
```

HW3.2 - Part 1 - Mapper

In [435]:
```
%%writefile mapper321.py
#!/usr/bin/python
## mapper321.py
## Author: James Gray
## Description: mapper code for HW3.2 Part 1

import sys
import re

# capture how many times Mapper is executed in a user-defined counter
sys.stderr.write("reporter:counter:Mapper Counter,Calls,1\n")

# input comes from STDNIN and is specified in the Hadoop Streaming job
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
    # create tokens
    words = line.split()

    # iterate through words
    for word in words:
        # create key-value pair for each word and send to STDOUT
        print('%s\t%s' % (word, 1))
```

```
Overwriting mapper321.py
```

HW3.2 - Part 1 - Reducer

Code based on example from Michael G. Noll [http://www.michael-noll.com/tutorials/writing-an-hadoop-mapreduce-program-in-python/ (http://www.michael-noll.com/tutorials/writing-an-hadoop-mapreduce-program-in-python/)](http://www.michael-noll.com/tutorials/writing-an-hadoop-mapreduce-program-in-python/)

In [436]:
```
%%writefile reducer321.py
#!/usr/bin/python
## reducer321.py
## Author: James Gray
## Description: reducer code for HW3.2 Part 1

from operator import itemgetter
import sys

# capture how many times Reducer is executed in a user-defined counter
sys.stderr.write("reporter:counter:Reducer Counter,Calls,1\n")

current_word = None
current_count = 0
word = None

# input comes from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()

    # parse the input we got from mapper.py
    word, count = line.split('\t', 1)

    # convert count (currently a string) to int
    try:
        count = int(count)
    except ValueError:
        # count was not a number, so silently
        # ignore/discard this line
        continue

    # this IF-switch only works because Hadoop sorts map output
    # by key (here: word) before it is passed to the reducer
    if current_word == word:
        current_count += count
    else:
        if current_word:
            # write result to STDOUT
            print '%s\t%s' % (current_word, current_count)
        current_count = count
        current_word = word

# do not forget to output the last word if needed!
if current_word == word:
```

```
Overwriting reducer321.py
```

HW3.1 - Part 1 - Execute MR Job

In [437]:
```
# delete output directory and files
!hdfs dfs -rm -r /user/graymatter/hw31_job1-output

!hadoop jar /usr/local/Cellar/hadoop/2.7.1/libexec/share/hadoop/tools/l
-D mapred.map.tasks=2 \
-D mapred.reduce.tasks=4 \
-files mapper321.py,reducer321.py -mapper mapper321.py -reducer reducer
```

```
bytes from disk
16/05/30 14:59:43 INFO reduce.MergeManagerImpl: Merging 0 segments,
0 bytes from memory into reduce
16/05/30 14:59:43 INFO mapred.Merger: Merging 1 sorted segments
16/05/30 14:59:43 INFO mapred.Merger: Down to the last merge-pass, w
ith 1 segments left of total size: 4 bytes
16/05/30 14:59:43 INFO mapred.LocalJobRunner: 1 / 1 copied.
16/05/30 14:59:43 INFO streaming.PipeMapRed: PipeMapRed exec [/User
s/jamesgray/OneDrive/GitHub/W261_MachineLearningAtScale/HW03/./reduc
er321.py]
16/05/30 14:59:43 INFO streaming.PipeMapRed: R/W/S=1/0/0 in:NA [rec/
s] out:NA [rec/s]
16/05/30 14:59:43 INFO streaming.PipeMapRed: Records R/W=1/1
16/05/30 14:59:43 INFO streaming.PipeMapRed: MRErrorThread done
16/05/30 14:59:43 INFO streaming.PipeMapRed: mapRedFinished
16/05/30 14:59:43 INFO mapred.Task: Task:attempt_local1725054756_000
1_r_000002_0 is done. And is in the process of committing
16/05/30 14:59:43 INFO mapred.LocalJobRunner: 1 / 1 copied.
16/05/30 14:59:43 INFO mapred.Task: Task attempt_local1725054756_000
1_r_000002_0 is allowed to commit now
```

HW3.1 - Part 1 - Output Analysis

The output confirmed that the Mapper was called "1" time and the Reducer was called "4" times

```
                Failed Shuffles=0
                Merged Map outputs=4
                GC time elapsed (ms)=0
                Total committed heap usage (bytes)=1082654720
        Mapper
                Calls=1
        Reducer
                Calls=4
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=30
        File Output Format Counters
                Bytes Written=26
16/05/27 09:01:00 INFO streaming.StreamJob: Output directory: /user
```

In [50]:
```
!hdfs dfs -cat /user/grraumatter/hw21_job1_output/*
```

```
16/05/27 09:20:58 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
quux    2
foo     3
bar     1
labs    1
```

HW3.2 - Part 2 - Mapper

In [438]:
```
%%writefile mapper322.py
#!/usr/bin/python
## mapper321.py
## Author: James Gray
## Description: mapper code for HW3.2 Part 2

import sys
import re
from csv import reader

# capture how many times Mapper is executed in a user-defined counter
sys.stderr.write("reporter:counter:Mapper Counter,Calls,1\n")

WORD_RE = re.compile(r"[\w']+")

# input comes from STDNIN and is specified in the Hadoop Streaming job
for line in reader(sys.stdin): # returns list of strings
    # create tokens
    issues=re.findall(WORD_RE, line[3]) #extract issue (4th field)

    # iterate through issues and produce KV pairs
    for issue in issues:
        # create key-value pair for each word and send to STDOUT
```

```
Overwriting mapper322.py
```

In [65]:
```
!cat Consumer_Complaints.csv | ./mapper322.py > test.txt
```

```
reporter:counter:Mapper,Calls,1
```

HW3.2 - Part 2 - Reducer

In [439]:
```
%%writefile reducer322.py
#!/usr/bin/python
## reducer321.py
## Author: James Gray
## Description: reducer code for HW3.2 Part 2

from operator import itemgetter
import sys

# capture how many times Reducer is executed in a user-defined counter
sys.stderr.write("reporter:counter:Reducer Counter,Calls,1\n")

current_word = None
current_count = 0
word = None

# input comes from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()

    # parse the input we got from mapper.py
    word, count = line.split('\t', 1)

    # convert count (currently a string) to int
    try:
        count = int(count)
    except ValueError:
        # count was not a number, so silently
        # ignore/discard this line
        continue

    # this IF-switch only works because Hadoop sorts map output
    # by key (here: word) before it is passed to the reducer
    if current_word == word:
        current_count += count
    else:
        if current_word:
            # write result to STDOUT
            print '%s\t%s' % (current_word, current_count)
        current_count = count
        current_word = word

# do not forget to output the last word if needed!
if current_word == word:
```

```
Overwriting reducer322.py
```

In [66]:
```
# delete output directory
!hdfs dfs -rm -r /user/graymatter/hw322-output

!hadoop jar /usr/local/Cellar/hadoop/2.7.1/libexec/share/hadoop/tools/l
-D mapred.map.tasks=2 \
-D mapred.reduce.tasks=2 \
-files mapper322.py,reducer322.py -mapper mapper322.py -reducer reducer
```

```
16/05/27 10:14:55 INFO mapreduce.Job: Job job_local712592919_0001 ru
nning in uber mode : false
16/05/27 10:14:55 INFO mapreduce.Job:  map 0% reduce 0%
16/05/27 10:14:56 INFO streaming.PipeMapRed: R/W/S=100000/448120/0 i
n:100000=100000/1 [rec/s] out:448120=448120/1 [rec/s]
16/05/27 10:14:57 INFO streaming.PipeMapRed: R/W/S=200000/882972/0 i
n:100000=200000/2 [rec/s] out:441486=882972/2 [rec/s]
16/05/27 10:14:58 INFO streaming.PipeMapRed: R/W/S=300000/1294842/0
in:100000=300000/3 [rec/s] out:431614=1294842/3 [rec/s]
16/05/27 10:14:58 INFO streaming.PipeMapRed: MRErrorThread done
16/05/27 10:14:58 INFO streaming.PipeMapRed: mapRedFinished
16/05/27 10:14:58 INFO mapred.LocalJobRunner:
16/05/27 10:14:58 INFO mapred.MapTask: Starting flush of map output
16/05/27 10:14:58 INFO mapred.MapTask: Spilling map output
16/05/27 10:14:58 INFO mapred.MapTask: bufstart = 0; bufend = 134247
15; bufvoid = 104857600
16/05/27 10:14:58 INFO mapred.MapTask: kvstart = 26214396(10485758
4); kvend = 20821164(83284656); length = 5393233/6553600
16/05/27 10:14:59 INFO mapred.MapTask: Finished spill 0
16/05/27 10:14:59 INFO mapred.Task: Task:attempt_local712592919_0001
```

HW3.2 - Part 2 - Output

```
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=10
                Total committed heap usage (bytes)=1117782016
    Mapper
                Calls=1
    Reducer
                Calls=2
    Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
```

In [68]:
```
!hdfs dfs -cat /user/graymatter/hw323-output/*
```

```
16/05/27 10:18:09 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
APR     3431
Account 16555
Applied 139
Arbitration     168
Bankruptcy      222
Billing 8158
Can't   1999
Cash    240
Closing 2795
Cont'd  17972
Convenience     75
Credit  14768
Debt    1343
Delinquent      1061
Deposits        10555
Disclosure      7655
False   3621
```

HW3.2 - Part 3 - Mapper

This mapper is identical to the original Mapper in Part 1

In [69]:
```
%%writefile mapper323.py
#!/usr/bin/python
## mapper321.py
## Author: James Gray
## Description: mapper code for HW3.2 Part 3

import sys
import re
from csv import reader

# capture how many times Mapper is executed in a user-defined counter
sys.stderr.write("reporter:counter:Mapper Counter,Calls,1\n")

WORD_RE = re.compile(r"[\w']+")

# input comes from STDNIN and is specified in the Hadoop Streaming job
for line in reader(sys.stdin): # returns list of strings
    # create tokens
    issues=re.findall(WORD_RE, line[3]) #extract issue (4th field)

    # iterate through issues and produce KV pairs
    for issue in issues:
        # create key-value pair for each word and send to STDOUT
        print('%s\t%s' % (issue, 1))
```

```
Overwriting mapper323.py
```

HW3.2 - Part 3 - Combiner

In this part we will use a combiner which will aggregate the word frequencies in the mapper and emit key-value pair for the unique word and count. Combiners actually act as "mini-reducers" that process the output of mappers (page 42 - MapReduce Algorithm Design). So in this scenario we will use the "reducer" code that performs the aggregation as the combiner.

In [440]:
```
%%writefile combiner323.py
#!/usr/bin/python
## combiner321.py
## Author: James Gray
## Description: combiner code for HW3.2 Part 3

from operator import itemgetter
import sys

# capture how many times Reducer is executed in a user-defined counter
sys.stderr.write("reporter:counter:Combiner Counter,Calls,1\n")

current_word = None
current_count = 0
word = None

# input comes from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()

    # parse the input we got from mapper.py
    word, count = line.split('\t', 1)

    # convert count (currently a string) to int
    try:
        count = int(count)
    except ValueError:
        # count was not a number, so silently
        # ignore/discard this line
        continue

    # this IF-switch only works because Hadoop sorts map output
    # by key (here: word) before it is passed to the reducer
    if current_word == word:
        current_count += count
    else:
        if current_word:
            # write result to STDOUT
            print '%s\t%s' % (current_word, current_count)
        current_count = count
        current_word = word

# do not forget to output the last word if needed!
if current_word == word:
```

```
Overwriting combiner323.py
```

HW3.2 - Part 3 - Reducer

In [441]:
```
%%writefile reducer323.py
#!/usr/bin/python
## reducer323.py
## Author: James Gray
## Description: reducer code for HW3.2 Part 3

from operator import itemgetter
import sys

# capture how many times Reducer is executed in a user-defined counter
sys.stderr.write("reporter:counter:Reducer Counter,Calls,1\n")

current_word = None
current_count = 0
word = None

# input comes from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()

    # parse the input we got from mapper.py
    word, count = line.split('\t', 1)

    # convert count (currently a string) to int
    try:
        count = int(count)
    except ValueError:
        # count was not a number, so silently
        # ignore/discard this line
        continue

    # this IF-switch only works because Hadoop sorts map output
    # by key (here: word) before it is passed to the reducer
    if current_word == word:
        current_count += count
    else:
        if current_word:
            # write result to STDOUT
            print '%s\t%s' % (current_word, current_count)
        current_count = count
        current_word = word

# do not forget to output the last word if needed!
if current_word == word:
```

```
Overwriting reducer323.py
```

In [43]:

```
# set file priveleges to execute script
!chmod a+x reducer323.py
!chmod a+x mapper323.py
```

HW3.2 - Part 3 - Execute MR Job

In [70]:
```
# delete output directory
!hdfs dfs -rm -r /user/graymatter/hw323-output

!hadoop jar /usr/local/Cellar/hadoop/2.7.1/libexec/share/hadoop/tools/l
-D mapred.map.tasks=2 \
-D mapred.reduce.tasks=2 \
-files mapper323.py,reducer323.py,combiner323.py -mapper mapper323.py -
```

```
16/05/27 10:20:25 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
16/05/27 10:20:25 INFO fs.TrashPolicyDefault: Namenode trash configu
ration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted /user/graymatter/hw323-output
16/05/27 10:20:26 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
16/05/27 10:20:27 INFO Configuration.deprecation: session.id is depr
ecated. Instead, use dfs.metrics.session-id
16/05/27 10:20:27 INFO jvm.JvmMetrics: Initializing JVM Metrics with
processName=JobTracker, sessionId=
16/05/27 10:20:27 INFO jvm.JvmMetrics: Cannot initialize JVM Metrics
with processName=JobTracker, sessionId= - already initialized
16/05/27 10:20:27 INFO mapred.FileInputFormat: Total input paths to
process : 1
16/05/27 10:20:27 INFO mapreduce.JobSubmitter: number of splits:1
16/05/27 10:20:27 INFO Configuration.deprecation: mapred.map.tasks i
```

HW3.2 - Part 3 - Output Summar

```
        Total committed heap usage (bytes)=986710016
    Combiner
            Calls=2
    Mapper
            Calls=1
    Reducer
            Calls=2
    Shuffle Errors
            BAD_ID=0
            CONNECTION=0
            IO_ERROR=0
```

In [71]:
```
16/05/27 10:20:54 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
APR     3431
Account 16555
Applied 139
Arbitration     168
Bankruptcy      222
Billing 8158
Can't   1999
Cash    240
Closing 2795
Cont'd  17972
Convenience     75
Credit  14768
Debt    1343
Delinquent      1061
Deposits        10555
Disclosure      7655
```

HW3.2 - Part 4

Using a single reducer:

What are the top 50 most frequent terms in your word count analysis? Present the top 50 terms and their frequency and their relative frequency.

If there are ties please sort the tokens in alphanumeric/string order. Present bottom 10 tokens (least frequent items).

In [128]:
```
%%writefile mapper324.py
#!/usr/bin/python
## mapper324.py
## Author: James Gray
## Description: mapper code for HW3.2 Part 4

import sys
import re
from csv import reader

word_total = 0

# capture how many times Mapper is executed in a user-defined counter
sys.stderr.write("reporter:counter:Mapper Counter,Calls,1\n")

WORD_RE = re.compile(r"[\w']+")

# input comes from STDNIN and is specified in the Hadoop Streaming job
for line in reader(sys.stdin): # returns list of strings
    # create tokens
    issues=re.findall(WORD_RE, line[3]) #extract issue (4th field)

    # iterate through issues and produce KV pairs
    for issue in issues:
        word_total+=1
        # create key-value pair for each word and send to STDOUT
        print('%s\t%s') % (issue.lower(), 1)

# emit total number of words for calculating relative frequency
print('*WORD_TOTAL' + '\t' + str(word_total))
```

```
Overwriting mapper324.py
```

In [131]:

```
%%writefile reducer324.py
#!/usr/bin/python
## reducer324.py
## Author: James Gray
## Description: reducer code for HW3.2 Part 4

from __future__ import division
from operator import itemgetter

import sys

# capture how many times Reducer is executed in a user-defined counter
sys.stderr.write("reporter:counter:Reducer Counter,Calls,1\n")

current_word = None
current_count = 0
word = None
word_total = 0

# input comes from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()

    # parse the input we got from mapper.py
    word, count = line.split('\t', 1)

    # convert count (currently a string) to int
    try:
        count = int(count)
    except ValueError:
        # count was not a number, so silently
        # ignore/discard this line
        continue

    # total the words to calculate a relative frequency
    if word == "*WORD.TOTAL":
        word_total = count

    # this IF-switch only works because Hadoop sorts map output
    # by key (here: word) before it is passed to the reducer
    if current_word == word:
        current_count += count
    else:
        if current_word:
            # write result to STDOUT
            word_relative = current_count/word_total
            print '%s\t%s\t%s' % (current_word, str(current_count), str
        current_count = count
        current_word = word

# do not forget to output the last word if needed!
if current_word == word:
    #print '%s\t%s' % (current_word, current_count)
```

```
    # reverse the output so we can sort on the current_count
    word_relative = current_count/word_total
```

```
Overwriting reducer324.py
```

In [132]:
```
# set file priveleges to execute script
!chmod a+x reducer324.py
```

In [129]:
```
reporter:counter:Mapper,Calls,1
```

In [147]:
```
# delete output directory
!hdfs dfs -rm -r /user/graymatter/hw324-output

!hadoop jar /usr/local/Cellar/hadoop/2.7.1/libexec/share/hadoop/tools/l
-D mapred.map.tasks=2 \
-D mapred.reduce.tasks=1 \
-files mapper324.py,reducer324.py -mapper mapper324.py -reducer reducer
```

```
16/05/27 22:06:44 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
16/05/27 22:06:44 INFO fs.TrashPolicyDefault: Namenode trash configu
ration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted /user/graymatter/hw324-output
16/05/27 22:06:46 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
16/05/27 22:06:47 INFO Configuration.deprecation: session.id is depr
ecated. Instead, use dfs.metrics.session-id
16/05/27 22:06:47 INFO jvm.JvmMetrics: Initializing JVM Metrics with
processName=JobTracker, sessionId=
16/05/27 22:06:47 INFO jvm.JvmMetrics: Cannot initialize JVM Metrics
with processName=JobTracker, sessionId= - already initialized
16/05/27 22:06:47 INFO mapred.FileInputFormat: Total input paths to
process : 1
16/05/27 22:06:47 INFO mapreduce.JobSubmitter: number of splits:1
16/05/27 22:06:47 INFO Configuration.deprecation: mapred.map.tasks i
```

In [134]:
```
!hdfs dfs -cat /user/graymatter/hw224-output/*
```

```
16/05/27 20:23:48 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
*WORD.TOTAL     1348309 1.0
a       3503    0.00259806913697
account 57448   0.0426074438426
acct    163     0.000120892169377
action  2964    0.0021983091413
advance 240     0.000178000740186
advertising     1193    0.000884812012677
amount  98      7.26836355761e-05
amt     71      5.26585523051e-05
an      2964    0.0021983091413
and     16448   0.0121989840608
application     8868    0.00657712734989
applied 139     0.000103092095358
apply   118     8.75170305917e-05
apr     3431    0.00254466891491
arbitration     168     0.00012460051813
```

In [137]:
```
%%writefile identity.py
#!/usr/bin/python

import sys
for line in sys.stdin:
    print line.strip()
```

```
Writing identity.py
```

In [144]:
```
!chmod +x identity.py
```

H3.2 - Part 4 - MR Job with Sorting

In [149]:
```
!hdfs dfs -rm -r user/graymatter/hw324-output-sorted

!hadoop jar /usr/local/Cellar/hadoop/2.7.1/libexec/share/hadoop/tools/l
 -D mapred.map.tasks=2 \
 -D mapred.reduce.tasks=1 \
 -D stream.num.map.output.key.fields=2 \ # word and value
 -D mapred.output.key.comparator.class=org.apache.hadoop.mapred.lib.Key
 -D  mapred.text.key.comparator.options='-k2,2nr' \ #sort on 2nd field(
 -file ./identity.py    -mapper ./identity.py \
  -reducer ./identity.py \
 -input /user/graymatter/hw324-output -output /user/graymatter/hw324_ou
```

```
16/05/27 22:09:02 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
rm: `user/graymatter/hw324-output-sorted': No such file or directory
16/05/27 22:09:03 WARN streaming.StreamJob: -file option is deprecat
ed, please use generic option -files instead.
16/05/27 22:09:04 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
packageJobJar: [./identity.py] [] /var/folders/ld/9wpyxfw13t7_pdv_0b
8958x40000gn/T/streamjob3744883328114669627.jar tmpDir=null
16/05/27 22:09:05 INFO Configuration.deprecation: session.id is depr
ecated. Instead, use dfs.metrics.session-id
16/05/27 22:09:05 INFO jvm.JvmMetrics: Initializing JVM Metrics with
processName=JobTracker, sessionId=
16/05/27 22:09:05 INFO jvm.JvmMetrics: Cannot initialize JVM Metrics
with processName=JobTracker, sessionId= - already initialized
16/05/27 22:09:05 INFO mapred.FileInputFormat: Total input paths to
process : 1
16/05/27 22:09:05 INFO mapreduce.JobSubmitter: number of splits:1
```

In [151]:
```
!hdfs dfs -cat /user/graymatter/hw324-output-sorted/*
```

```
16/05/27 22:11:41 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
*WORD.TOTAL     1348309 1.0
loan    119630  0.0887259522854
collection      72394   0.0536924399377
foreclosure     70487   0.052278075723
modification    70487   0.052278075723
account 57448   0.0426074438426
credit  55251   0.0409779954002
or      40508   0.0300435582645
payments        39993   0.0296615983428
servicing       36767   0.0272689717268
escrow  36767   0.0272689717268
report  34903   0.0258864993114
incorrect       29133   0.0216070648494
information     29069   0.0215595979853
on      29069   0.0215595979853
debt    27874   0.0206733026332
```

In [166]:
```
# Generate top 50 words
print("Top 50 words - word, frequency, relative frequency")
!hdfs dfs -cat /user/graymatter/hw324_output-sorted/* | head -50
print("")

# Generate least common 10 words
print("Least common 10 words - word, frequency, relative frequency")
!hdfs dfs -cat /user/graymatter/hw324_output-sorted/* | tail -10
```

```
Top 50 words - word, frequency, relative frequency
16/05/28 13:39:56 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
*WORD.TOTAL     1348309 1.0
loan    119630  0.0887259522854
collection      72394   0.0536924399377
foreclosure     70487   0.052278075723
modification    70487   0.052278075723
account 57448   0.0426074438426
credit  55251   0.0409779954002
or      40508   0.0300435582645
payments        39993   0.0296615983428
servicing       36767   0.0272689717268
escrow  36767   0.0272689717268
report  34903   0.0258864993114
incorrect       29133   0.0216070648494
information     29069   0.0215595979853
on      29069   0.0215595979853
debt    27874   0.0206733026332
closing 19000   0.0140917252648
not     18477   0.0137038319851
attempts        17972   0.013329288761
owed    17972   0.013329288761
collect 17972   0.013329288761
cont'd  17972   0.013329288761
and     16448   0.0121989840608
opening 16205   0.0120187583113
management      16205   0.0120187583113
of      13983   0.0103707681251
my      10731   0.00795885809558
withdrawals     10555   0.00782832421945
deposits        10555   0.00782832421945
problems        9484    0.00703399591637
application     8868    0.00657712734989
tactics 8671    0.00643101840898
communication   8671    0.00643101840898
mortgage        8625    0.00639690160045
originator      8625    0.00639690160045
broker  8625    0.00639690160045
to      8401    0.00623076757628
unable  8178    0.00606537522185
billing 8158    0.00605054182684
other   7886    0.00584880765463
verification    7655    0.0056774819422
disclosure      7655    0.0056774819422
disputes        6938    0.00514570473089
reporting       6559    0.00486461189534
lease   6337    0.00469996121067
the     6248    0.00463395260285
by      5663    0.00420007579865
caused  5663    0.00420007579865
funds   5663    0.00420007579865
being   5663    0.00420007579865
```

```
Least common 10 words - word, frequency, relative frequency
16/05/28 13:39:57 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
apply   118     8.75170305917e-05
amount  98      7.26836355761e-05
payment 92      6.82336170715e-05
credited        92      6.82336170715e-05
checks  75      5.56252313083e-05
convenience     75      5.56252313083e-05
amt     71      5.26585523051e-05
day     71      5.26585523051e-05
disclosures     64      4.74668640497e-05
missing 64      4.74668640497e-05
```

HW3.2.1

Using 2 reducers: What are the top 50 most frequent terms in your word count analysis? Present the top 50 terms and their frequency and their relative frequency. If there are ties please sort the tokens in alphanumeric/string order. Present bottom 10 tokens (least frequent items). Please use a combiner.

HW3.2.1 Mapper

Since we have two reducers, we will emit KV pairs that will get routed to two partitioners

In [509]:
```
%%writefile mapper3215.py
#!/usr/bin/python
## mapper3215.py
## Author: James Gray
## Description: mapper code for HW3.2.1

import sys
import re
from csv import reader

word_total = 0

# Create two partitioners for the reducers
partition1 = "abcdefghijklm"
partition2 = "nopqrstuvwxyz"

# capture how many times Mapper is executed in a user-defined counter
sys.stderr.write("reporter:counter:Mapper Counter,Calls,1\n")

WORD_RE = re.compile(r"[\w']+")

# input comes from STDNIN and is specified in the Hadoop Streaming job
for line in reader(sys.stdin): # returns list of strings
    # create tokens
    issues=re.findall(WORD_RE, line[3]) #extract issue (4th field)

    # iterate through issues and produce KV pairs
    for issue in issues:
        word_total+=1
        if issue[0].lower() in partition1:
            partitionKey = "a"
        else:
            partitionKey = "b"

        # create key-value pair for each word and send to STDOUT
        print('%s\t%s\t%s') % (issue.lower(), 1, partitionKey)

# emit total number of words for calculating relative frequency
```

```
Overwriting mapper3215.py
```

HW3.2.1 Reducer

In [510]:

```
%%writefile reducer3215.py
#!/usr/bin/python
## reducer3215.py
## Author: James Gray
## Description: reducer code for HW3.2.1

from __future__ import division
from operator import itemgetter

import sys

# capture how many times Reducer is executed in a user-defined counter
sys.stderr.write("reporter:counter:Reducer Counter,Calls,1\n")

current_word = None
current_count = 0
word = None
word_total = 0

# input comes from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()

    # parse the input we got from mapper.py
    word, count, partition = line.split('\t')

    # convert count (currently a string) to int
    try:
        count = int(count)
    except ValueError:
        # count was not a number, so silently
        # ignore/discard this line
        continue

    # total the words to calculate a relative frequency
    if word == "*WORD.TOTAL":
        word_total = count

    # this IF-switch only works because Hadoop sorts map output
    # by key (here: word) before it is passed to the reducer
    if current_word == word:
        current_count += count
    else:
        if current_word:
            # write result to STDOUT
            word_relative = current_count/word_total
            print '%s\t%s\t%s' % (current_word, str(current_count), str
        current_count = count
        current_word = word

# do not forget to output the last word if needed!
if current_word == word:
    #print '%s\t%s' % (current_word, current_count)
```

```
    # reverse the output so we can sort on the current_count
    word_relative = current_count/word_total
```

```
Overwriting reducer3215.py
```

In [511]:
```
!chmod +x mapper3215.py
!chmod +x reducer3215.py
```

In [512]:
```
!cat Consumer_Complaints.csv | ./mapper3215.py > test3215.txt
```

```
reporter:counter:Mapper Counter,Calls,1
```

In [513]:
```
# delete output directory
!hdfs dfs -rm -r /user/graymatter/hw3215-output

!hadoop jar /usr/local/Cellar/hadoop/2.7.1/libexec/share/hadoop/tools/l
-D mapred.output.key.comparator.class=org.apache.hadoop.mapred.lib.KeyF
-D stream.map.output.field.separator="\t" \
-D mapreduce.partition.keypartitioner.options="-k3,3" \
-D mapreduce.partition.keycomparator.options="-k2,2nr" \
-D mapred.map.tasks=1 \
-D mapred.reduce.tasks=2 \
-files mapper3215.py,reducer3215.py -mapper mapper3215.py -reducer redu
-input /user/graymatter/consumer_complaints.csv -output /user/graymatte
-partitioner org.apache.hadoop.mapred.lib.KeyFieldBasedPartitioner \
```

```
  File "<ipython-input-513-9a01b722bb0f>", line 5
    -D mapreduce.partition.keypartitioner.options="-k3,3" \
               ^
SyntaxError: invalid syntax
```

In [497]:
```
!hdfs dfs -cat /user/graymatter/hw3215-output/*
```

```
16/05/31 08:37:15 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
1348309 1348309 1.0
```

HW3.3 - Shopping Cart Analysis

Product Recommendations: The action or practice of selling additional products or services to existing customers is called cross-selling. Giving product recommendation is one of the examples of cross-selling that are frequently used by online retailers. One simple method to give product recommendations is to recommend products that are frequently browsed together by the customers.

For this homework use the online browsing behavior dataset located at:

```
    https://www.dropbox.com/s/zlfyiwa70poqg74/ProductPurchaseDa
ta.txt?dl=0
```

Each line in this dataset represents a browsing session of a customer. On each line, each string of 8 characters represents the id of an item browsed during that session. The items are separated by spaces.

Here are the first few lines of the ProductPurchaseData
FRO11987 ELE17451 ELE89019 SNA90258 GRO99222 GRO99222 GRO12298 FRO12685 ELE91550 SNA11465 ELE26917 ELE52966 FRO90334 SNA30755 ELE17451 FRO84225 SNA80192 ELE17451 GRO73461 DAI22896 SNA99873 FRO86643 ELE17451 ELE37798 FRO86643 GRO56989 ELE23393 SNA11465 ELE17451 SNA69641 FRO86643 FRO78087 SNA11465 GRO39357 ELE28573 ELE11375 DAI54444

Do some exploratory data analysis of this dataset guided by the following questions:

- How many unique items are available from this supplier?

Using a single reducer:

- Report your findings such as number of unique products;
- largest basket;
- report the top 50 most frequently purchased items, their frequency, and their relative frequency (break ties by sorting the products alphabetical order) etc. using Hadoop Map-Reduce.

In [442]:
```
# Load text file into HDFS

#!hdfs dfs -put ProductPurchaseData.txt /user/graymatter/ProductPurchas

!hdfs dfs -ls /user/graymatter/ProductPurchaseData.txt
```

```
16/05/30 15:02:17 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
-rw-r--r--   1 jamesgray supergroup    3458517 2016-05-28 14:59 /use
r/graymatter/ProductPurchaseData.txt
```

HW3.3 - Mapper

Given the calculations that are required we must:

- produce KV pairs that include the products but also by basket since there is question regarding the largest basket size.
- count the total number of products to calculate relative frequency.
- count the number of products by basket to determine the largest basket size

In [443]:
```
%%writefile mapper33.py
#!/usr/bin/python
## mapper33.py
## Author: James Gray
## Description: reducer code for HW3.3

import sys
basket_number = 0
total_products = 0

# capture how many times Mapper is executed in a user-defined counter
sys.stderr.write("reporter:counter:Mapper Counter,Calls,1\n")

for line in sys.stdin:
    line = line.strip()
    products = line.split() # each line is split into products using a

    # create basket ID
    basket_id = "basket_" + str(basket_number)

    for product in products:
        total_products+=1 # increment to calculate total number of prod
        # emit K-V pairs: product \t 1 \t basket_size \t basket_id
        print('%s\t%s\t%s\t%s') % (product, 1, str(len(products)), bask

    # increment basket number
    basket_number+=1

# emit total number of products across all baskets
print('%s\t%s\t%s\t%s') % ("*TOTAL_PRODUCTS", str(total_products), 0, 0
```

```
Overwriting mapper33.py
```

In [444]:

```
%%writefile reducer33.py
#!/usr/bin/python
## reducer33.py
## Author: James Gray
## Description: reducer code for HW3.3

from __future__ import division
from operator import itemgetter

import sys

# capture how many times Reducer is executed in a user-defined counter
sys.stderr.write("reporter:counter:Reducer,Calls,1\n")

current_product = None
current_count = 0
product = None
total_products = 0
unique_products = 0
baskets={} # dict to hold basket_id, basket_size
prod_relative = 0.0

# input comes from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()

    # parse the input we got from mapper.py
    product, count, basket_size, basket_id = line.split('\t')

    #convert strings into integers
    count = int(count)
    basket_size = int(basket_size)

    # total the words to calculate a relative frequency
    # this should be the first row of the mapper output file as its sor
    if product == "*TOTAL.PRODUCTS":
        total_products = count

    # keep track of the basket_id, basket_size
    baskets[basket_id] = {'size': basket_size}

    # this IF-switch only works because Hadoop sorts map output
    # by key (here: word) before it is passed to the reducer
    if current_product == product:
        current_count += count
    else:
        if current_product:
            # write result to STDOUT
            prod_relative = current_count/total_products
            print '%s\t%s\t%s' % (current_product, str(current_count),
            unique_products+=1
        current_count = count
        current_product = product
```

```
# do not forget to output the last word if needed!
if current_product == product:
    #print '%s\t%s' % (current_word, current_count)
    # reverse the output so we can sort on the current_count
    prod_relative = current_count/total_products
    print '%s\t%s\t%s' % (current_product, str(current_count), str(prod
    unique_products+=1

# emit the largest basket size
largest_basket = max(baskets, key=baskets.get)
largest_basketsize = baskets[largest_basket]['size']
print("LARGEST.BASKET\t" + largest_basket + '\t' + str(largest_basketsi

# emit the number of unique products
```

```
Overwriting reducer33.py
```

In [445]:
```
!chmod +x mapper33.py
```

In [446]:
```
reporter:counter:Mapper Counter,Calls,1
```

In [447]:
```
# delete output directory
!hdfs dfs -rm -r /user/graymatter/hw33-output

!hadoop jar /usr/local/Cellar/hadoop/2.7.1/libexec/share/hadoop/tools/l
-D mapred.map.tasks=2 \
-D mapred.reduce.tasks=1 \
-files mapper33.py,reducer33.py -mapper mapper33.py -reducer reducer33.
```

```
er33.py]
16/05/30 15:36:53 INFO Configuration.deprecation: mapred.job.tracker
is deprecated. Instead, use mapreduce.jobtracker.address
16/05/30 15:36:53 INFO Configuration.deprecation: mapred.map.tasks i
s deprecated. Instead, use mapreduce.job.maps
16/05/30 15:36:53 INFO streaming.PipeMapRed: R/W/S=1/0/0 in:NA [rec/
s] out:NA [rec/s]
16/05/30 15:36:53 INFO streaming.PipeMapRed: R/W/S=10/0/0 in:NA [re
c/s] out:NA [rec/s]
16/05/30 15:36:53 INFO streaming.PipeMapRed: R/W/S=100/0/0 in:NA [re
c/s] out:NA [rec/s]
16/05/30 15:36:53 INFO streaming.PipeMapRed: R/W/S=1000/0/0 in:NA [r
ec/s] out:NA [rec/s]
16/05/30 15:36:53 INFO mapreduce.Job:  map 100% reduce 0%
16/05/30 15:36:53 INFO streaming.PipeMapRed: R/W/S=10000/0/0 in:NA
[rec/s] out:NA [rec/s]
16/05/30 15:36:53 INFO streaming.PipeMapRed: Records R/W=19838/1
16/05/30 15:36:53 INFO streaming.PipeMapRed: R/W/S=100000/2237/0 in:
NA [rec/s] out:NA [rec/s]
```

In [242]:
```
!hdfs dfs -cat /user/graymatter/hw33-output/*
```

```
16/05/29 11:09:36 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
*TOTAL.PRODUCTS 380824  1.0
DAI11153        8       2.10070793858e-05
DAI11223        155     0.000407012163099
DAI11238        3       7.87765476966e-06
DAI11257        1       2.62588492322e-06
DAI11261        6       1.57553095393e-05
DAI11273        1       2.62588492322e-06
DAI11290        5       1.31294246161e-05
DAI11299        2       5.25176984644e-06
DAI11375        1       2.62588492322e-06
DAI11462        8       2.10070793858e-05
DAI11541        5       1.31294246161e-05
DAI11552        8       2.10070793858e-05
DAI11555        25      6.56471230805e-05
DAI11582        1       2.62588492322e-06
DAI11613        2       5.25176984644e-06
```

HW3.3. Run MR Job with Sort

In [243]:
```
# delete output directory
!hdfs dfs -rm -r /user/graymatter/hw33-outputsorted

!hadoop jar /usr/local/Cellar/hadoop/2.7.1/libexec/share/hadoop/tools/l
-D mapred.map.tasks=2 \
-D mapred.reduce.tasks=1 \
-D stream.num.map.output.key.fields=2 \
-D mapred.output.key.comparator.class=org.apache.hadoop.mapred.lib.KeyF
-D mapred.text.key.comparator.options='-k2,2nr -k1,1' \
-file identity.py -mapper identity.py -reducer identity.py \
```

```
16/05/29 11:09:46 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
rm: `/user/graymatter/hw33-outputsorted': No such file or directory
16/05/29 11:09:48 WARN streaming.StreamJob: -file option is deprecat
ed, please use generic option -files instead.
16/05/29 11:09:48 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
packageJobJar: [identity.py] [] /var/folders/ld/9wpyxfw13t7_pdv_0b89
58x40000gn/T/streamjob1149202741170728182.jar tmpDir=null
16/05/29 11:09:49 INFO Configuration.deprecation: session.id is depr
ecated. Instead, use dfs.metrics.session-id
16/05/29 11:09:49 INFO jvm.JvmMetrics: Initializing JVM Metrics with
processName=JobTracker, sessionId=
16/05/29 11:09:49 INFO jvm.JvmMetrics: Cannot initialize JVM Metrics
with processName=JobTracker, sessionId= - already initialized
16/05/29 11:09:49 INFO mapred.FileInputFormat: Total input paths to
process : 1
```

In [245]:
```
!hdfs dfs -cat /user/graymatter/hw33-outputsorted/*
```

```
16/05/29 11:10:34 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
*TOTAL.PRODUCTS 380824  1.0
UNIQUE.PRODUCTS 12593
DAI62779        6667    0.0175067747831
FRO40251        3881    0.010191059387
ELE17451        3875    0.0101753040775
GRO73461        3602    0.00945843749344
SNA80324        3044    0.00799319370628
ELE32164        2851    0.0074863979161
DAI75645        2736    0.00718442114993
SNA45677        2455    0.0064465474865
FRO31317        2330    0.0061183118711
DAI85309        2293    0.00602115412894
ELE26917        2292    0.00601852824402
FRO80039        2233    0.00586360103355
GRO21487        2115    0.00555374661261
SNA99873        2083    0.00546971829507
```

Top 50 products -> product | purchase frequency | relative frequency

In [247]:
```
!hdfs dfs -cat /user/graymatter/hw33-outputsorted/* | head -52
```

```
16/05/29 11:15:18 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
*TOTAL.PRODUCTS 380824  1.0
UNIQUE.PRODUCTS 12593
DAI62779        6667    0.0175067747831
FRO40251        3881    0.010191059387
ELE17451        3875    0.0101753040775
GRO73461        3602    0.00945843749344
SNA80324        3044    0.00799319370628
ELE32164        2851    0.0074863979161
DAI75645        2736    0.00718442114993
SNA45677        2455    0.0064465474865
FRO31317        2330    0.0061183118711
DAI85309        2293    0.00602115412894
ELE26917        2292    0.00601852824402
FRO80039        2233    0.00586360103355
GRO21487        2115    0.00555374661261
SNA99873        2083    0.00546971829507
GRO59710        2004    0.00526227338613
GRO71621        1920    0.00504169905258
FRO85978        1918    0.00503644728273
GRO30386        1840    0.00483162825872
ELE74009        1816    0.00476860702057
GRO56726        1784    0.00468457870302
DAI63921        1773    0.00465569396887
GRO46854        1756    0.00461105392517
ELE66600        1713    0.00449814087347
DAI83733        1712    0.00449551498855
FRO32293        1702    0.00446925613932
ELE66810        1697    0.0044561267147
SNA55762        1646    0.00432220658362
DAI22177        1627    0.00427231477008
FRO78087        1531    0.00402022981745
ELE99737        1516    0.0039808415436
ELE34057        1489    0.00390994265067
GRO94758        1489    0.00390994265067
FRO35904        1436    0.00377077074974
FRO53271        1420    0.00372875659097
SNA93860        1407    0.00369462008697
SNA90094        1390    0.00364998004327
GRO38814        1352    0.00355019641619
ELE56788        1345    0.00353181522173
GRO61133        1321    0.00346879398357
DAI88807        1316    0.00345566455896
ELE74482        1316    0.00345566455896
ELE59935        1311    0.00344253513434
SNA96271        1295    0.00340052097557
DAI43223        1290    0.00338739155095
ELE91337        1289    0.00338476566603
GRO15017        1275    0.0033480032771
DAI31081        1261    0.00331124088818
GRO81087        1220    0.00320357960633
DAI22896        1219    0.0032009537214
```

```
GRO85051        1214    0.00318782429679
cat: Unable to write to output stream.
```

HW3.3 - Output Summary

- Total number of unique products = 12,593
- Total number of products sold = 380,824
- Largest basket = 37 products in basket_id 7033

HW3.3.1 OPTIONAL

Using 2 reducers: Report your findings such as number of unique products; largest basket; report the top 50 most frequently purchased items, their frequency, and their relative frequency (break ties by sorting the products alphabetical order) etc. using Hadoop Map-Reduce.

HW3.4 - (Computationally prohibitive but then again Hadoop can handle this) Pairs

Suppose we want to recommend new products to the customer based on the products they have already browsed on the online website. Write a map-reduce program to find products which are frequently browsed together. Fix the support count (cooccurence count) to s = 100 (i.e. product pairs need to occur together at least 100 times to be considered frequent) and find pairs of items (sometimes referred to itemsets of size 2 in association rule mining) that have a support count of 100 or more.

List the top 50 product pairs with corresponding support count (aka frequency), and relative frequency or support (number of records where they coccur, the number of records where they coccur/the number of baskets in the dataset) in decreasing order of support for frequent (100>count) itemsets of size 2.

Use the Pairs pattern (lecture 3) to extract these frequent itemsets of size 2. Free free to use combiners if they bring value. Instrument your code with counters for count the number of times your mapper, combiner and reducers are called.

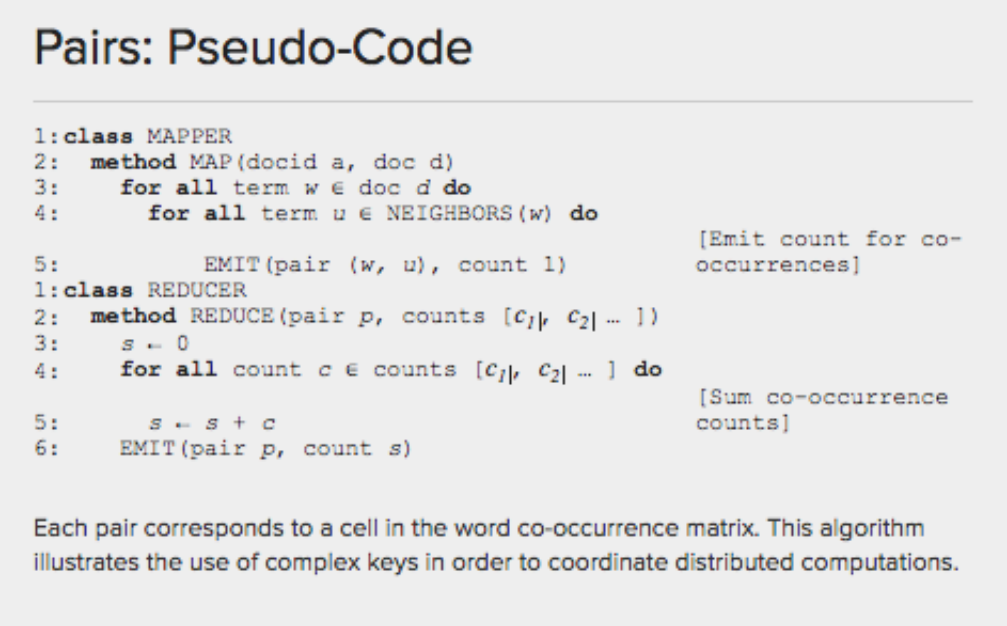
Please output records of the following form for the top 50 pairs (itemsets of size 2):

```
item1, item2, support count, support
```

Fix the ordering of the pairs lexicographically (left to right), and break ties in support (between pairs, if any exist) by taking the first ones in lexicographically increasing order.

Report the compute time for the Pairs job. Describe the computational setup used (E.g., single computer; dual core; linux, number of mappers, number of reducers)Instrument your mapper, combiner, and reducer to count how many times each is called using Counters and report these counts.

Pairs Design Pattern Pseudocode from Async Week 3



Pairs: Pseudo-Code

```
1:class MAPPER
2:  method MAP(docid a, doc d)
3:    for all term w ∈ doc d do
4:      for all term u ∈ NEIGHBORS(w) do
                                              [Emit count for co-
5:          EMIT(pair (w, u), count 1)        occurrences]
1:class REDUCER
2:  method REDUCE(pair p, counts [c1|, c2| … ])
3:    s ← 0
4:    for all count c ∈ counts [c1|, c2| … ] do
                                              [Sum co-occurrence
5:      s ← s + c                             counts]
6:    EMIT(pair p, count s)
```

Each pair corresponds to a cell in the word co-occurrence matrix. This algorithm illustrates the use of complex keys in order to coordinate distributed computations.

HW3.4 Mapper

This mapper will implement the pseudo-code above by emitting

In [316]:
```
%%writefile mapper34.py
#!/usr/bin/python
## mapper34.py
## Author: James Gray
## Description: reducer code for HW3.4

import sys
baskets_total = 0 #calculate the total number of baskets in the dataset

# capture how many times Mapper is executed in a user-defined counter
sys.stderr.write("reporter:counter:Mapper,Calls,1\n")

total_pairs=0

for line in sys.stdin:
    line = line.strip()
    products = line.split() # each line is split into products using a

    # count the total number of baskets
    baskets_total+=1

    # iterate through each product in basket from left to right
    for i,product in enumerate(products): # this will return a tuple of

        current_product = product

        # now index to the next products in the basket after the curren
        for next_product in products[i+1:]: #select the subset of produ

            # product a consistent itemset format by sorting the pairs
            itemset = sorted([current_product, next_product])

            # increment itemset pair count
            total_pairs+=1

            # emit K-V pairs: product \t 1 \t basket_size \t basket_id
            #print('%s\t%s\t%s') % (itemset[0],itemset[1], 1)
            print('%s\t%s') % (itemset[0]+','+itemset[1], 1)

# emit total number of baskets using "*" for order inversion to enable
#print('%s\t%s') % ("*TOTAL.BASKETS", str(baskets_total))
print('%s\t%s') % ("*TOTAL.PAIRS", str(total_pairs))
```

```
Overwriting mapper34.py
```

HW3.4 Reducer

In [317]:

```
%%writefile reducer34.py
#!/usr/bin/python
## reducer34.py
## Author: James Gray
## Description: reducer code for HW3.4

from __future__ import division # this needs to be the first line of cc
import sys

# capture how many times Mapper is executed in a user-defined counter
sys.stderr.write("reporter:counter:Reducer,Calls,1\n")

s=100 # frequency of product pairs to be considered frequent

current_itemset = None

current_count = 0
product = None
total_products = 0
unique_products = 0

prod_relative = 0.0

# input comes from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()

    # parse the input we got from mapper.py
    itemset, count = line.split('\t')

    #convert strings into integers
    count = int(count)

    # total the words to calculate a relative frequency
    # this should be the first row of the mapper output file as its sor
    if itemset == "*TOTAL.PAIRS":
        total_products = count

    # keep track of the basket_id, basket_size
    #baskets[basket_id] = {'size': basket_size}

    # this IF-switch only works because Hadoop sorts map output
    # by key (here: word) before it is passed to the reducer
    if current_itemset == itemset:
        current_count += count
    else:
        if current_itemset and current_count>=s:
            # write result to STDOUT
            prod_relative = current_count/total_products

            if current_itemset!= "*TOTAL.PAIRS":
                print '%s\t%s\t%s' % (current_itemset, str(current_coun
            #unique_products+=1
```

```
        current_count = count
        current_itemset = itemset

# do not forget to output the last word if needed!
if current_itemset and current_count>=s:

    prod_relative = current_count/total_products

    if current_itemset!= "*TOTAL.PAIRS":
        print '%s\t%s\t%s' % (current_itemset, str(current_count), str(
    #unique_products+=1

# emit the largest basket size
#largest_basket = max(baskets, key=baskets.get)
#largest_basketsize = baskets[largest_basket]['size']
#print("LARGEST.BASKET\t" + largest_basket + '\t' + str(largest_baskets

# emit the number of unique products
#print("UNIQUE.PRODUCTS\t" + str(unique_products))
```

```
Overwriting reducer34.py
```

In [318]:
```
!chmod +x mapper34.py
!chmod +x reducer34.py
```

In [425]:
```
#!cat ProductPurchaseData.txt | ./mapper34.py | sort > output34.txt

!cat output34.txt | ./reducer34.py | sort -k2,2nr > output34final.txt |
```

```
reporter:counter:Reducer,Calls,1
```

In [319]:
```
# delete output directory
!hdfs dfs -rm -r /user/graymatter/hw34-output

!hadoop jar /usr/local/Cellar/hadoop/2.7.1/libexec/share/hadoop/tools/l
-D mapred.map.tasks=2 \
-D mapred.reduce.tasks=1 \
-files mapper34.py,reducer34.py -mapper mapper34.py -reducer reducer34.
```

```
16/05/29 20:04:42 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
16/05/29 20:04:43 INFO fs.TrashPolicyDefault: Namenode trash configu
ration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted /user/graymatter/hw34-output
16/05/29 20:04:44 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
16/05/29 20:04:45 INFO Configuration.deprecation: session.id is depr
ecated. Instead, use dfs.metrics.session-id
16/05/29 20:04:45 INFO jvm.JvmMetrics: Initializing JVM Metrics with
processName=JobTracker, sessionId=
16/05/29 20:04:45 INFO jvm.JvmMetrics: Cannot initialize JVM Metrics
with processName=JobTracker, sessionId= - already initialized
16/05/29 20:04:45 INFO mapred.FileInputFormat: Total input paths to
process : 1
16/05/29 20:04:46 INFO mapreduce.JobSubmitter: number of splits:1
16/05/29 20:04:46 INFO Configuration.deprecation: mapred.map.tasks i
```

In [ ]:

In [399]:
```
!hdfs dfs -cat /user/graymatter/hw34-output/*
```

```
DAI22896,ELE17451       193     7.616245412e-05
DAI22896,ELE32164       107     4.22247802634e-05
DAI22896,ELE74009       165     6.51129789109e-05
DAI22896,FRO31317       167     6.59022271401e-05
DAI22896,FRO40251       154     6.07721136502e-05
DAI22896,FRO53271       123     4.85387660972e-05
DAI22896,GRO21487       114     4.49871490657e-05
DAI22896,GRO30386       102     4.02516596904e-05
DAI22896,GRO38814       223     8.80011775584e-05
DAI22896,GRO46854       114     4.49871490657e-05
DAI22896,GRO61133       110     4.34086526073e-05
DAI22896,GRO73461       304     0.000119965730842
DAI22896,SNA72163       227     8.95796740168e-05
DAI22896,SNA80324       195     7.69517023492e-05
DAI23334,DAI62779       273     0.000107732383289
DAI23334,ELE17451       100     3.94624114611e-05
DAI23334,ELE92920       143     5.64312483894e-05
DAI29159,DAI62779       119     4.69602696388e-05
DAI31081,DAI43223       123     4.85387660972e-05
DAI31081,DAI62779       364     0.000143643177719
```

In [320]:
```
!hdfs dfs -rm -r /user/graymatter/hw34-output-sorted

!hadoop jar /usr/local/Cellar/hadoop/2.7.1/libexec/share/hadoop/tools/l
-D mapred.map.tasks=2 \
-D mapred.reduce.tasks=1 \
-D stream.num.map.output.key.fields=2 \
-D mapred.output.key.comparator.class=org.apache.hadoop.mapred.lib.KeyF
-D mapred.text.key.comparator.options='-k2,2nr' \
-files identity.py -mapper identity.py -reducer identity.py \
```

```
s/jamesgray/OneDrive/GitHub/W261_MachineLearningAtScale/HW03/./ident
ity.py]
16/05/29 20:05:51 INFO Configuration.deprecation: mapred.work.outpu
t.dir is deprecated. Instead, use mapreduce.task.output.dir
16/05/29 20:05:51 INFO Configuration.deprecation: map.input.start is
deprecated. Instead, use mapreduce.map.input.start
16/05/29 20:05:51 INFO Configuration.deprecation: mapred.task.is.map
is deprecated. Instead, use mapreduce.task.ismap
16/05/29 20:05:51 INFO Configuration.deprecation: mapred.task.id is
deprecated. Instead, use mapreduce.task.attempt.id
16/05/29 20:05:51 INFO Configuration.deprecation: mapred.tip.id is d
eprecated. Instead, use mapreduce.task.id
16/05/29 20:05:51 INFO Configuration.deprecation: mapred.local.dir i
s deprecated. Instead, use mapreduce.cluster.local.dir
16/05/29 20:05:51 INFO Configuration.deprecation: map.input.file is
deprecated. Instead, use mapreduce.map.input.file
16/05/29 20:05:51 INFO Configuration.deprecation: mapred.skip.on is
deprecated. Instead, use mapreduce.job.skiprecords
16/05/29 20:05:51 INFO Configuration.deprecation: map.input.length i
s deprecated. Instead, use mapreduce.map.input.length
```

In [313]:
```
!hdfs dfs -cat /user/graymatter/hw34-output-sorted/*
```

```
16/05/29 19:46:31 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
DAI62779,ELE17451       1592    0.000628241590461
FRO40251,SNA80324       1412    0.000557209249831
DAI75645,FRO40251       1254    0.000494858639723
FRO40251,GRO85051       1213    0.000478679051024
DAI62779,GRO73461       1139    0.000449476866542
DAI75645,SNA80324       1130    0.000445925249511
DAI62779,FRO40251       1070    0.000422247802634
DAI62779,SNA80324       923     0.000364238057786
DAI62779,DAI85309       918     0.000362264937213
ELE32164,GRO59710       911     0.000359502568411
FRO40251,GRO73461       882     0.000348058469087
DAI62779,DAI75645       882     0.000348058469087
DAI62779,ELE92920       877     0.000346085348514
FRO40251,FRO92469       835     0.000329511135701
DAI62779,ELE32164       832     0.000328327263357
DAI75645,GRO73461       712     0.000280972369603
DAI42223,ELE32164       711     0.000280577745490
```

In [314]:
```
!hdfs dfs -cat /user/graymatter/hw34_output_sorted/* | head -50
```

```
16/05/29 19:46:39 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
DAI62779,ELE17451       1592    0.000628241590461
FRO40251,SNA80324       1412    0.000557209249831
DAI75645,FRO40251       1254    0.000494858639723
FRO40251,GRO85051       1213    0.000478679051024
DAI62779,GRO73461       1139    0.000449476866542
DAI75645,SNA80324       1130    0.000445925249511
DAI62779,FRO40251       1070    0.000422247802634
DAI62779,SNA80324       923     0.000364238057786
DAI62779,DAI85309       918     0.000362264937213
ELE32164,GRO59710       911     0.000359502568411
FRO40251,GRO73461       882     0.000348058469087
DAI62779,DAI75645       882     0.000348058469087
DAI62779,ELE92920       877     0.000346085348514
FRO40251,FRO92469       835     0.000329511135701
DAI62779,ELE32164       832     0.000328327263357
DAI75645,GRO73461       712     0.000280972369603
DAI43223,ELE32164       711     0.000280577745489
DAI62779,GRO30386       709     0.00027978849726
ELE17451,FRO40251       697     0.000275053007884
DAI85309,ELE99737       659     0.000260057291529
DAI62779,ELE26917       650     0.000256505674497
GRO21487,GRO73461       631     0.00024900781632
DAI62779,SNA45677       604     0.000238352965225
ELE17451,SNA80324       597     0.000235590596423
DAI62779,GRO71621       595     0.000234801348194
DAI62779,SNA55762       593     0.000234012099965
DAI62779,DAI83733       586     0.000231249731162
ELE17451,GRO73461       580     0.000228881986475
GRO73461,SNA80324       562     0.000221778752412
DAI62779,GRO59710       561     0.000221384128297
DAI62779,FRO80039       550     0.000217043263036
DAI75645,ELE17451       547     0.000215859390692
DAI62779,SNA93860       537     0.000211913149546
DAI55148,DAI62779       526     0.000207572284286
DAI43223,GRO59710       512     0.000202047546681
ELE17451,ELE32164       511     0.000201652922566
DAI62779,SNA18336       506     0.000199679801993
ELE32164,GRO73461       486     0.000191787319701
DAI85309,ELE17451       482     0.000190208823243
DAI62779,FRO78087       482     0.000190208823243
DAI62779,GRO94758       479     0.000189024950899
DAI62779,GRO21487       471     0.000185867957982
GRO85051,SNA80324       471     0.000185867957982
ELE17451,GRO30386       468     0.000184684085638
FRO85978,SNA95666       463     0.000182710965065
DAI62779,FRO19221       462     0.00018231634095
DAI62779,GRO46854       461     0.000181921716836
DAI43223,DAI62779       459     0.000181132468607
ELE92920,SNA18336       455     0.000179553972148
DAI88079,FRO40251       446     0.000176002355117
cat: Unable to write to output stream.
```

In [315]:
```
!hdfs dfs -cat /user/graymatter/hw34_output_sorted/* | tail -10
```

```
16/05/29 19:48:10 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
ELE14480,SNA80324       100     3.94624114611e-05
ELE17451,ELE37770       100     3.94624114611e-05
GRO46854,SNA66583       100     3.94624114611e-05
GRO73461,GRO88511       100     3.94624114611e-05
FRO80039,GRO64900       100     3.94624114611e-05
DAI62779,GRO17075       100     3.94624114611e-05
FRO78087,GRO94758       100     3.94624114611e-05
FRO78087,GRO30386       100     3.94624114611e-05
DAI63921,ELE11160       100     3.94624114611e-05
GRO38814,SNA93860       100     3.94624114611e-05
```

HW3.4 Conclusions

```
This code was run on a MacBook with 16GB RAM, 2cores, 3.1GHz.

First MR job with 2 mappers(ran 2x), 1 reducer (ran 1x) ran in 27
seconds
Second MR job with 2 mappers(ran 1x, 1 reducer (ran 1x) ran in 4
seconds
```

HW3.5: Stripes

Repeat 3.4 using the stripes design pattern for finding cooccuring pairs.

- Report the compute times for stripes job versus the Pairs job.
- Describe the computational setup used (E.g., single computer; dual core; linux, number of mappers, number of reducers)
- Instrument your mapper, combiner, and reducer to count how many times each is called using Counters and report these counts. Discuss the differences in these counts between the Pairs and Stripes jobs

Stripes: Pseudo-Code

a → [b: 1, c: 2, d: 5, e: 3, f: 2]

```
1:  class MAPPER
2:    method MAP(docid a, doc d)
3:      for all term u ∈ doc d do
4:      H ← new ASSOCIATIVEARRAY
5:      for all term u ∈ NEIGHBORS (w) do
6:        H{u} ← H{u} + 1            Tally words co-occurring with w
7:      EMIT(Term w, Stripe H
```

```
1:  class REDUCER
2:    method REDUCE(term w, stripes [H1, H2, H3, ... ])
3:      Hf ← new ASSOCIATIVEARRAY
4:      for all stripe H ∈ stripes [H1, H2, H3] do
5:          SUM(Hf, H)              Element-wise sum
6:        EMIT(term w, stripe Hf)
```

- **Idea: group together pairs into an associative array**

  (a, b) → 1
  (a, c) → 2
  (a, d) → 5 a → { b: 1, c: 2, d: 5, e: 3, f: 2 }
  (a, e) → 3
  (a, f) → 2

- **Each mapper takes a sentence:**
  - Generate all co-occurring term pairs
  - For each term, emit a → { b: $count_b$, c: $count_c$, d: $count_d$ … }
- **Reducers perform element-wise sum of associative arrays**

```
    a → { b: 1,         d: 5, e: 3 }
+   a → { b: 1, c: 2, d: 2,        f: 2 }
    a → { b: 2, c: 2, d: 7, e: 3, f: 2 }
```

Key: cleverly-constructed data structure brings together partial results

Large-Scale Machine Learning, MIDS, UC Berkeley © 2015 James G. Shanahan Contact: James.Shanahan @ gmail.com 249

HW3.5 Mapper

The mapper will emit a KV-pair based on the design pattern above (product \t associative array).

In [427]:
```
%%writefile mapper35.py
#!/usr/bin/python
## mapper35.py
## Author: James Gray
## Description: reducer code for HW3.4

import sys

# capture how many times Mapper is executed in a user-defined counter
sys.stderr.write("reporter:counter:Mapper Counter,Calls,1\n")

total_pairs=0

for line in sys.stdin:
    line = line.strip()
    products = line.split() # each line is split into products using a

    try:
        products.sort() # get all products left to right alphabetical
    except:
        pass

    # count the total number of baskets
    #baskets_total+=1

    # iterate through each product in basket from left to right
    for i,product in enumerate(products): # this will return a tuple of

        current_product = product

        stripes = {} # use dictionary for associative array

        # now index to the next products in the basket after the curren
        for next_product in products[i+1:]: #select the subset of produ

            # increment itemset pair count
            total_pairs+=1

            # build associative array for current product
            try:
                # increment count if product appears in basket more tha
                stripes[next_product]+=1
            except KeyError:
                # add next_product to dictionary for current product
                stripes[next_product]=1

        # emit K-V pairs: current product \t stripes
        print('%s\t%s') % (current_product, str(stripes))

# emit total number of baskets using "*" for order inversion to enable
print('%s\t%s') % ("*TOTAL_PAIRS", str(total_pairs))
```

```
Overwriting mapper35.py
```

HW3.5 Reducer

This reducer will add the associative arrays with the same product key

In [413]:

```
%%writefile reducer35.py
#!/usr/bin/python
## reducer35.py
## Author: James Gray
## Description: reducer code for HW3.5

from __future__ import division # this needs to be the first line of cc
import sys
from collections import Counter

# capture how many times Reducer is executed in a user-defined counter
sys.stderr.write("reporter:counter:Reducer Counter,Calls,1\n")

s=100 # frequency of product pairs to be considered frequent

current_product = None

current_count = 0
product = None
total_products = 0
unique_products = 0
final_counter = Counter({})
#prod_dict ={}

prod_relative = 0.0

# input comes from STDIN
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()

    # parse the input we got from mapper.py
    product, prod_dict = line.split('\t')

    # total the words to calculate a relative frequency
    # this should be the first row of the mapper output file as its sor
    if product == "*TOTAL.PAIRS":
        total_products = int(prod_dict)

    # convert dicionary string back to dictionary object
    if product != "*TOTAL.PAIRS":
        prod_dict = Counter(eval(prod_dict))

    # keep track of the basket_id, basket_size
    #baskets[basket_id] = {'size': basket_size}

    # this IF-switch only works because Hadoop sorts map output
    # by key (here: word) before it is passed to the reducer
    if current_product == product:
        # https://docs.python.org/3.5/library/collections.html#collecti
        # Two counters can be added and the dict values are added
        final_counter += Counter(prod_dict)
    else:
        if current_product and current_product!= "*TOTAL.PAIRS":
```

```
            # write result to STDOUT

            # iterate through each of the dictionary entries and check
            for key in final_counter:
                if final_counter[key] >= s:
                    print '%s\t%s\t%s\t%s' % (current_product, key, str
                                              str((final_counter[key]/t
        current_product=product
        final_counter = prod_dict #reset counter to current product

# do not forget to output the last word if needed!
if current_product and current_product!= "*TOTAL.PAIRS":

    # iterate through each of the dictionary entries and check if frequ
    for key in final_counter:
        if final_counter[key] >= s:
            print '%s\t%s\t%s\t%s' % (current_product, key, str(final_c
                                      str((final_counter[key]/total_pro

# emit the largest basket size
#largest_basket = max(baskets, key=baskets.get)
#largest_basketsize = baskets[largest_basket]['size']
#print("LARGEST.BASKET\t" + largest_basket + '\t' + str(largest_baskets

# emit the number of unique products
#print("UNIQUE.PRODUCTS\t" + str(unique_products))
```

```
Overwriting reducer35.py
```

In [414]:
```
!chmod +x mapper35.py
!chmod +x reducer35.py
```

HW3.5 - Execute MR Job on Unix

In [430]:
```
#!cat ProductPurchaseData.txt | ./mapper35.py | sort > output35.txt

!cat output35.txt | ./reducer35.py | sort -k3,3nr > output35final.txt
```

```
reporter:counter:Reducer,Calls,1
```

HW3.5 - MR Job #1

In [431]:
```
# delete output directory
!hdfs dfs -rm -r /user/graymatter/hw35-output

!hadoop jar /usr/local/Cellar/hadoop/2.7.1/libexec/share/hadoop/tools/l
-D mapred.map.tasks=2 \
-D mapred.reduce.tasks=1 \
-files mapper35.py,reducer35.py -mapper mapper35.py -reducer reducer35.
-input /user/graymatter/ProductPurchaseData.txt -output /user/graymatte
```

```
16/05/30 14:42:47 INFO mapred.LocalJobRunner: Records R/W=138198/411
> reduce
16/05/30 14:42:50 INFO mapred.LocalJobRunner: Records R/W=138198/411
> reduce
16/05/30 14:42:50 INFO mapreduce.Job:  map 100% reduce 87%
16/05/30 14:42:53 INFO mapred.LocalJobRunner: Records R/W=138198/411
> reduce
16/05/30 14:42:53 INFO mapreduce.Job:  map 100% reduce 88%
16/05/30 14:42:56 INFO mapred.LocalJobRunner: Records R/W=138198/411
> reduce
16/05/30 14:42:59 INFO mapred.LocalJobRunner: Records R/W=138198/411
> reduce
16/05/30 14:42:59 INFO mapreduce.Job:  map 100% reduce 89%
16/05/30 14:43:02 INFO mapred.LocalJobRunner: Records R/W=138198/411
> reduce
16/05/30 14:43:02 INFO mapreduce.Job:  map 100% reduce 90%
16/05/30 14:43:05 INFO mapred.LocalJobRunner: Records R/W=138198/411
> reduce
16/05/30 14:43:08 INFO mapred.LocalJobRunner: Records R/W=138198/411
> reduce
```

In [409]:
```
!hdfs dfs -cat /user/graymatter/hw35-output/* | head -50
```

```
16/05/30 13:33:32 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
DAI11778        DAI62779        169     6.66914753693e-05
DAI11778        SNA80324        120     4.73548937534e-05
DAI11778        DAI75645        128     5.05118866703e-05
DAI11927        GRO73461        122     4.81441419826e-05
DAI11927        GRO46854        155     6.11667377648e-05
DAI13194        FRO31077        286     0.000112862496779
DAI13194        GRO67376        252     9.94452768821e-05
DAI13266        GRO44993        131     5.16957590141e-05
DAI13788        DAI31081        102     4.02516596904e-05
DAI13788        FRO31317        111     4.38032767219e-05
DAI13788        GRO32086        140     5.52473760456e-05
DAI13902        FRO41069        110     4.34086526073e-05
DAI13902        DAI62779        281     0.000110889376206
DAI13902        ELE17451        162     6.39291065671e-05
DAI13902        SNA80324        110     4.34086526073e-05
DAI13902        ELE32164        180     7.10323406301e-05
DAI13902        SNA55952        135     5.32742554725e-05
DAI14125        FRO78087        188     7.4189333547e-05
DAI14125        ELE12845        199     7.85301988077e-05
DAI14125        FRO85978        110     4.34086526073e-05
DAI14125        ELE30911        172     6.78753477132e-05
DAI14125        DAI16732        159     6.27452342232e-05
DAI14470        GRO64900        239     9.43151633921e-05
DAI14902        FRO80039        104     4.10409079196e-05
DAI15595        DAI54142        184     7.26108370885e-05
DAI15595        DAI59441        131     5.16957590141e-05
DAI16142        GRO71621        103     4.0646283805e-05
DAI16142        DAI63921        143     5.64312483894e-05
DAI16142        DAI62779        328     0.000129436709593
DAI16142        GRO56726        125     4.93280143264e-05
DAI16142        GRO73461        172     6.78753477132e-05
DAI16142        ELE38536        232     9.15527945899e-05
DAI16142        GRO88505        101     3.98570355758e-05
DAI16142        GRO46854        127     5.01172625557e-05
DAI16732        ELE66600        213     8.40549364122e-05
DAI16732        FRO78087        396     0.000156271149386
DAI16732        DAI62779        153     6.03774895356e-05
DAI16732        GRO56726        116     4.57763972949e-05
DAI16732        GRO88324        103     4.0646283805e-05
DAI16732        ELE32164        114     4.49871490657e-05
DAI16732        ELE30911        151     5.95882413063e-05
DAI17368        DAI62779        100     3.94624114611e-05
DAI17810        ELE29795        189     7.45839576616e-05
DAI18527        FRO19221        110     4.34086526073e-05
DAI18527        FRO38071        122     4.81441419826e-05
DAI18527        GRO82670        187     7.37947094323e-05
DAI18527        DAI62779        113     4.45925249511e-05
DAI18527        ELE17451        111     4.38032767219e-05
DAI18527        FRO40251        102     4.02516596904e-05
DAI18527        DAI47529        121     4.7749517868e-05
cat: Unable to write to output stream.
```

HW3.5 - MR Job #2

In [432]:
```
!hdfs dfs -rm -r /user/graymatter/hw35-output-sorted

!hadoop jar /usr/local/Cellar/hadoop/2.7.1/libexec/share/hadoop/tools/l
-D mapred.map.tasks=2 \
-D mapred.reduce.tasks=1 \
-D stream.num.map.output.key.fields=3 \
-D mapred.output.key.comparator.class=org.apache.hadoop.mapred.lib.KeyF
-D mapred.text.key.comparator.options='-k3,3nr' \
-files identity.py -mapper identity.py -reducer identity.py \
```

```
16/05/30 14:45:29 INFO Configuration.deprecation: user.name is depre
cated. Instead, use mapreduce.job.user.name
16/05/30 14:45:29 INFO Configuration.deprecation: mapred.task.partit
ion is deprecated. Instead, use mapreduce.task.partition
16/05/30 14:45:29 INFO streaming.PipeMapRed: R/W/S=1/0/0 in:NA [rec/
s] out:NA [rec/s]
16/05/30 14:45:29 INFO streaming.PipeMapRed: R/W/S=10/0/0 in:NA [re
c/s] out:NA [rec/s]
16/05/30 14:45:29 INFO streaming.PipeMapRed: R/W/S=100/0/0 in:NA [re
c/s] out:NA [rec/s]
16/05/30 14:45:29 INFO streaming.PipeMapRed: R/W/S=1000/0/0 in:NA [r
ec/s] out:NA [rec/s]
16/05/30 14:45:29 INFO streaming.PipeMapRed: Records R/W=1334/1
16/05/30 14:45:29 INFO streaming.PipeMapRed: MRErrorThread done
16/05/30 14:45:29 INFO streaming.PipeMapRed: mapRedFinished
16/05/30 14:45:29 INFO mapred.LocalJobRunner:
16/05/30 14:45:29 INFO mapred.MapTask: Starting flush of map output
16/05/30 14:45:29 INFO mapred.MapTask: Spilling map output
16/05/30 14:45:29 INFO mapred.MapTask: bufstart = 0; bufend = 53238;
bufvoid = 104857600
```

In [433]:
```
!hdfs dfs -cat /user/graymatter/hw35_output_sorted/* | head -50
```

```
16/05/30 14:45:39 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
DAI62779        ELE17451        1592    0.000628241590461
FRO40251        SNA80324        1412    0.000557209249831
DAI75645        FRO40251        1254    0.000494858639723
FRO40251        GRO85051        1213    0.000478679051024
DAI62779        GRO73461        1139    0.000449476866542
DAI75645        SNA80324        1130    0.000445925249511
DAI62779        FRO40251        1070    0.000422247802634
DAI62779        SNA80324        923     0.000364238057786
DAI62779        DAI85309        918     0.000362264937213
ELE32164        GRO59710        911     0.000359502568411
DAI62779        DAI75645        882     0.000348058469087
FRO40251        GRO73461        882     0.000348058469087
DAI62779        ELE92920        877     0.000346085348514
FRO40251        FRO92469        835     0.000329511135701
DAI62779        ELE32164        832     0.000328327263357
DAI75645        GRO73461        712     0.000280972369603
DAI43223        ELE32164        711     0.000280577745489
DAI62779        GRO30386        709     0.00027978849726
ELE17451        FRO40251        697     0.000275053007884
DAI85309        ELE99737        659     0.000260057291529
DAI62779        ELE26917        650     0.000256505674497
GRO21487        GRO73461        631     0.00024900781632
DAI62779        SNA45677        604     0.000238352965225
ELE17451        SNA80324        597     0.000235590596423
DAI62779        GRO71621        595     0.000234801348194
DAI62779        SNA55762        593     0.000234012099965
DAI62779        DAI83733        586     0.000231249731162
ELE17451        GRO73461        580     0.000228881986475
GRO73461        SNA80324        562     0.000221778752412
DAI62779        GRO59710        561     0.000221384128297
DAI62779        FRO80039        550     0.000217043263036
DAI75645        ELE17451        547     0.000215859390692
DAI62779        SNA93860        537     0.000211913149546
DAI55148        DAI62779        526     0.000207572284286
DAI43223        GRO59710        512     0.000202047546681
ELE17451        ELE32164        511     0.000201652922566
DAI62779        SNA18336        506     0.000199679801993
ELE32164        GRO73461        486     0.000191787319701
DAI85309        ELE17451        482     0.000190208823243
DAI62779        FRO78087        482     0.000190208823243
DAI62779        GRO94758        479     0.000189024950899
DAI62779        GRO21487        471     0.000185867957982
GRO85051        SNA80324        471     0.000185867957982
ELE17451        GRO30386        468     0.000184684085638
FRO85978        SNA95666        463     0.000182710965065
DAI62779        FRO19221        462     0.00018231634095
DAI62779        GRO46854        461     0.000181921716836
DAI43223        DAI62779        459     0.000181132468607
ELE92920        SNA18336        455     0.000179553972148
DAI88079        FRO40251        446     0.000176002355117
```

In [434]:
```
!hdfs dfs -cat /user/graymatter/hw35_output_sorted/* | tail -10
```

```
16/05/30 14:46:26 WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java classes where
applicable
ELE17451        SNA59061        100     3.94624114611e-05
GRO59710        SNA93860        100     3.94624114611e-05
FRO80039        GRO64900        100     3.94624114611e-05
DAI85309        ELE14480        100     3.94624114611e-05
DAI62779        GRO17075        100     3.94624114611e-05
ELE17451        ELE37770        100     3.94624114611e-05
FRO78087        GRO94758        100     3.94624114611e-05
FRO78087        GRO30386        100     3.94624114611e-05
FRO40251        GRO56989        100     3.94624114611e-05
GRO38814        SNA93860        100     3.94624114611e-05
```

HW3.5 Conclusions

This code was run on a MacBook with 16GB RAM, 2cores, 3.1GHz.

- First MR job with 2 mappers(ran 2x), 1 reducer (ran 1x) ran in 27 seconds
- Second MR job with 2 mappers(ran 1x, 1 reducer (ran 1x) ran in 4 seconds

In [ ]:
```
```