**CSC 555 Mining Big Data**

Assignment 4

**Due Tuesday, March 5th**

1. Consider a Hadoop job that will result in 78 blocks of output to HDFS.

Suppose that writing an output block to HDFS takes 1 minute. The HDFS replication factor is set to 3 (for simplicity, we charge reducers for the cost of writing replicated blocks).

* 1. How long will it take for the reducer to write the job output on a 5-node Hadoop cluster? (ignoring the cost of Map processing, but counting replication cost in the output writing).

48 minutes

Calculation: 78/5 = 15.6, ceiling = 16 iterations

16 iterations x 1 minute = 16 minutes

16 minutes x 3 replication factor = 48 minutes

* 1. How long will it take for reducer(s) to write the job output to 10 Hadoop worker nodes? (Assume that data is distributed evenly and replication factor is set to 1)

8 minutes

Calculation: 78/10 = 7.8, ceiling = 8 iterations

8 iterations x 1 minute = 8 minutes

8 minutes x 1 replication factor = 8 minutes

* 1. How long will it take for reducer(s) to write the job output to 10 Hadoop worker nodes? (Assume that data is distributed evenly and replication factor is set to 3)

24 minutes

Calculation: 78/10 = 7.8, ceiling = 8 iterations

8 iterations x 1 minute = 8 minutes

8 minutes x 3 replication factor = 24 minutes

* 1. How long will it take for reducer(s) to write the job output to 100 Hadoop worker nodes? (Assume that data is distributed evenly and replication factor is set to 1)

1 minute

Calculation: 78/100 = .78, ceiling = 1 iteration

1 iteration x 1 minute = 1 minute

1 minute x 1 replication factor = 1 minute

* 1. How long will it take for reducer(s) to write the job output to 100 Hadoop worker nodes? (Assume that data is distributed evenly and replication factor is set to 3)

3 minutes

Calculation: 78/100 = .78, ceiling = 1 iteration

1 iteration x 1 minute = 1 minute

1 minute x 3 replication factor = 3 minutes

You can ignore the network transfer costs as well as the possibility of node failure.

1. 1. Consider the following graph

Compute the page rank for the nodes in this graph. If you are multiplying matrices manually, you may stop after 5 steps. If you use a tool (e.g., Matlab, website, etc.) for matrix multiplication, you should get your answer to converge.

The ranks converge after 28 iterations. The ranks are in the following table.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 28 | A | 0 | 1 | 1/2 | 1/2 | x | 4/11 | = | 4/11 |
| B | 0 | 0 | 1/2 | 0 | 1/11 | 1/11 |
| Y | 0 | 0 | 0 | 1/2 | 2/11 | 2/11 |
| Z | 1 | 0 | 0 | 0 | 361/993 | 4/11 |

The full set of calculations can be found at the end of this document.

* 1. Now consider a dead-end node Q and P:

What is the page rank of Q?

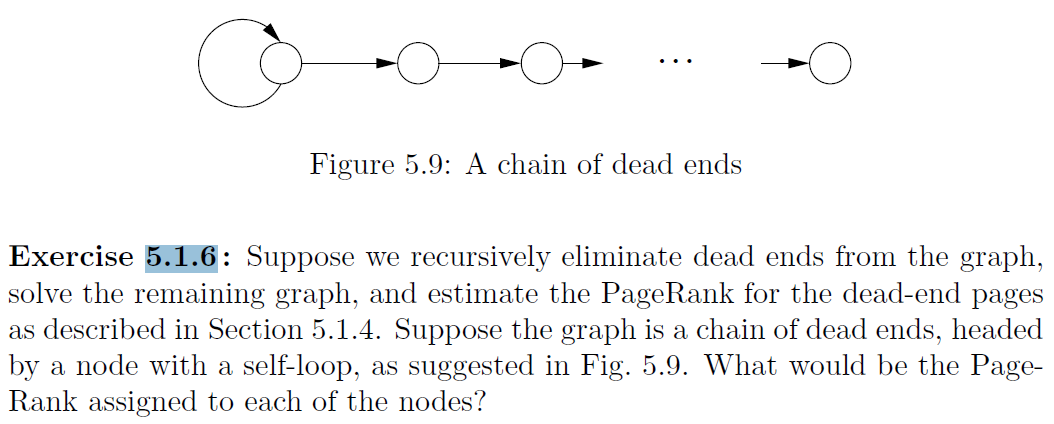
The calculation of the ranks for A, Y, and Z converge after 20 iterations. (Full calculation at the end of this document.)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | A | Y | Z |  | V |  | Rank |
| Step 20 | A | 0 | 1 | 1/2 | x | 2/5 | = | 2/5 |
| Y | 0 | 0 | 1/2 | 1/5 | 1/5 |
| Z | 1 | 0 | 0 | 307/768 | 2/5 |

Q has two predecessors, Y and Z. Y has two successors, so contributes 1/2 of its page rank to Q. Z has three successors, so contributes 1/3 of its page rank to Q. Q’s rank is calculated as:

1/2 x 1/5 + 1/3 x 2/5 = **7/30**

* 1. Exercise 5.1.6 from Mining of Massive Datasets



Eliminating all of the dead ends leaves only the root node. The root node has only a link to itself and thus has a rank of 1. The first dead-end node has only the root node as its predecessor. The root node has two successors, the first dead-end node and itself. Therefore, it contributes 1/2 of its rank to the first dead-end node. 1 x 1/2 = 1/2, so the first dead-end node’s rank is 1/2. All subsequent dead-end nodes are also ranked 1/2 because they all have only one predecessor and one successor.

1. Run another custom MapReduce job, implementing a solution for the following query:

For Employee(EID, EFirst, ELast, Phone) and Customer(CID, CFirst, CLast, Address), find everyone with the same name using MapReduce:

SELECT EFirst, ELast, Phone, Address

FROM Employee, Customer

WHERE EFirst = CFirst AND ELast = CLast;

You can use this input data:

http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/employee.txt

http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/customer.txt

Note that you can print all EID/Address values together (you do not have to print separate lines for each joined row that SQL query would output). Be sure to submit your python code, the command line and the screenshot of successful execution of your code.

**Mapper code (assignment4Mapper.py):**

#!/usr/bin/python

import sys

# input comes from STDIN (standard input)

for line in sys.stdin:

line = line.strip()

split = line.split('|')

if split[0].startswith('EMP'):

EFirst = split[1]

ELast = split[2]

Phone = split[3]

print EFirst, '\t', ELast, '\t', Phone, '\t', 'Employee'

else:

CFirst = split[1]

CLast = split[2]

Address = split[3]

print CFirst, '\t', CLast, '\t', Address, '\t', 'Customer'

**Reducer Code (assignment4Reducer.py):**

#!/usr/bin/python

import sys

currentKey = None

EFirst = None

ELast = None

Phone = None

Address = None

# input comes from STDIN

for line in sys.stdin:

split = line.strip().split('\t')

key = split[0] + ' ' + split[1]

value = '\t'.join(split[2:])

if currentKey == key: # Same key

if value.endswith('Employee'):

EFirst = split[0]

ELast = split[1]

Phone = split[2]

if value.endswith('Customer'):

Address = split[2]

else:

if currentKey:

lenPhone = len(Phone)

lenAddress = len(Address)

if (lenPhone\*lenAddress > 0):

# Join means that there have to be rows on each side

print EFirst, '\t', ELast, '\t', Phone, '\t', Address

EFirst = ''

ELast = ''

Phone = ''

Address = ''

currentKey = key

if value.endswith('Employee'):

EFirst = split[0]

ELast = split[1]

Phone = split[2]

Address = ''

elif value.endswith('Customer'):

EFirst = ''

ELast = ''

Phone = ''

Address = split[2]

lenPhone = len(Phone)

lenAddress = len(Address)

if (lenPhone\*lenAddress > 0):

# Join means that there have to be rows on each side

print EFirst, '\t', ELast, '\t', Phone, '\t', Address

lenPhone = len(Phone)

lenAddress = len(Address)

if (lenPhone\*lenAddress > 0):

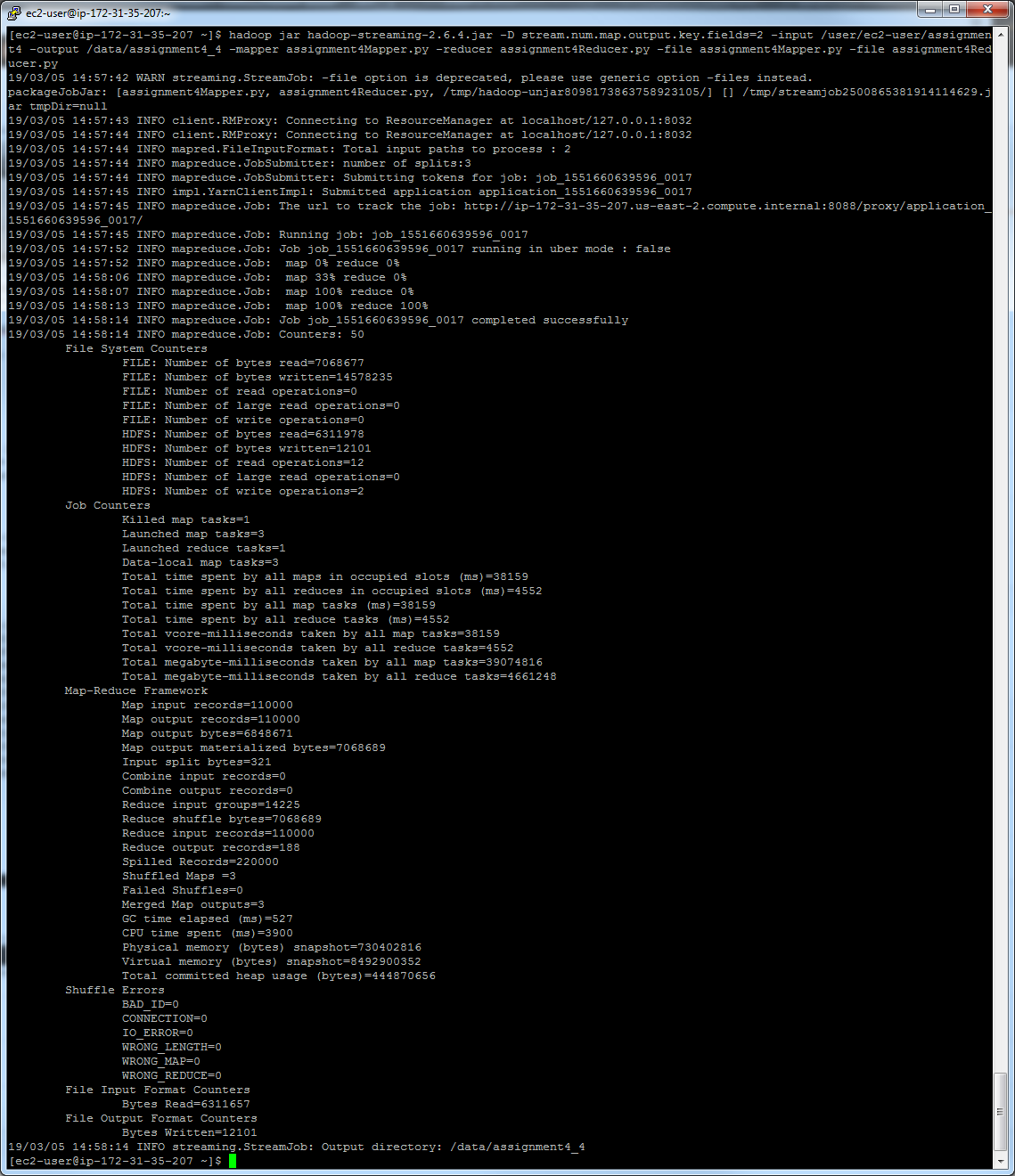
# Join means that there have to be rows on each side

print EFirst, '\t', 'ELast', '\t', Phone, '\t', Address

**Command:**

hadoop jar hadoop-streaming-2.6.4.jar -D stream.num.map.output.key.fields=2 -input /user/ec2-user/assignment4 -output /data/assignment4\_4 -mapper assignment4Mapper.py -reducer assignment4Reducer.py -file assignment4Mapper.py -file assignment4Reducer.py

**Execution output:**



**Full text:**

[ec2-user@ip-172-31-35-207 ~]$ hadoop jar hadoop-streaming-2.6.4.jar -D stream.num.map.output.key.fields=2 -input /user/ec2-user/assignment4 -output /data/assignment4\_4 -mapper assignment4Mapper.py -reducer assignment4Reducer.py -file assignment4Mapper.py -file assignment4Reducer.py

19/03/05 14:57:42 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.

packageJobJar: [assignment4Mapper.py, assignment4Reducer.py, /tmp/hadoop-unjar8098173863758923105/] [] /tmp/streamjob2500865381914114629.jar tmpDir=null

19/03/05 14:57:43 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032

19/03/05 14:57:44 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032

19/03/05 14:57:44 INFO mapred.FileInputFormat: Total input paths to process : 2

19/03/05 14:57:44 INFO mapreduce.JobSubmitter: number of splits:3

19/03/05 14:57:44 INFO mapreduce.JobSubmitter: Submitting tokens for job: job\_1551660639596\_0017

19/03/05 14:57:45 INFO impl.YarnClientImpl: Submitted application application\_1551660639596\_0017

19/03/05 14:57:45 INFO mapreduce.Job: The url to track the job: http://ip-172-31-35-207.us-east-2.compute.internal:8088/proxy/application\_1551660639596\_0017/

19/03/05 14:57:45 INFO mapreduce.Job: Running job: job\_1551660639596\_0017

19/03/05 14:57:52 INFO mapreduce.Job: Job job\_1551660639596\_0017 running in uber mode : false

19/03/05 14:57:52 INFO mapreduce.Job: map 0% reduce 0%

19/03/05 14:58:06 INFO mapreduce.Job: map 33% reduce 0%

19/03/05 14:58:07 INFO mapreduce.Job: map 100% reduce 0%

19/03/05 14:58:13 INFO mapreduce.Job: map 100% reduce 100%

19/03/05 14:58:14 INFO mapreduce.Job: Job job\_1551660639596\_0017 completed successfully

19/03/05 14:58:14 INFO mapreduce.Job: Counters: 50

File System Counters

FILE: Number of bytes read=7068677

FILE: Number of bytes written=14578235

FILE: Number of read operations=0

FILE: Number of large read operations=0

FILE: Number of write operations=0

HDFS: Number of bytes read=6311978

HDFS: Number of bytes written=12101

HDFS: Number of read operations=12

HDFS: Number of large read operations=0

HDFS: Number of write operations=2

Job Counters

Killed map tasks=1

Launched map tasks=3

Launched reduce tasks=1

Data-local map tasks=3

Total time spent by all maps in occupied slots (ms)=38159

Total time spent by all reduces in occupied slots (ms)=4552

Total time spent by all map tasks (ms)=38159

Total time spent by all reduce tasks (ms)=4552

Total vcore-milliseconds taken by all map tasks=38159

Total vcore-milliseconds taken by all reduce tasks=4552

Total megabyte-milliseconds taken by all map tasks=39074816

Total megabyte-milliseconds taken by all reduce tasks=4661248

Map-Reduce Framework

Map input records=110000

Map output records=110000

Map output bytes=6848671

Map output materialized bytes=7068689

Input split bytes=321

Combine input records=0

Combine output records=0

Reduce input groups=14225

Reduce shuffle bytes=7068689

Reduce input records=110000

Reduce output records=188

Spilled Records=220000

Shuffled Maps =3

Failed Shuffles=0

Merged Map outputs=3

GC time elapsed (ms)=527

CPU time spent (ms)=3900

Physical memory (bytes) snapshot=730402816

Virtual memory (bytes) snapshot=8492900352

Total committed heap usage (bytes)=444870656

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=6311657

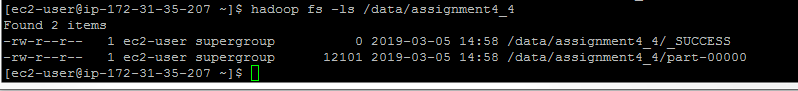
File Output Format Counters

Bytes Written=12101

19/03/05 14:58:14 INFO streaming.StreamJob: Output directory: /data/assignment4\_4

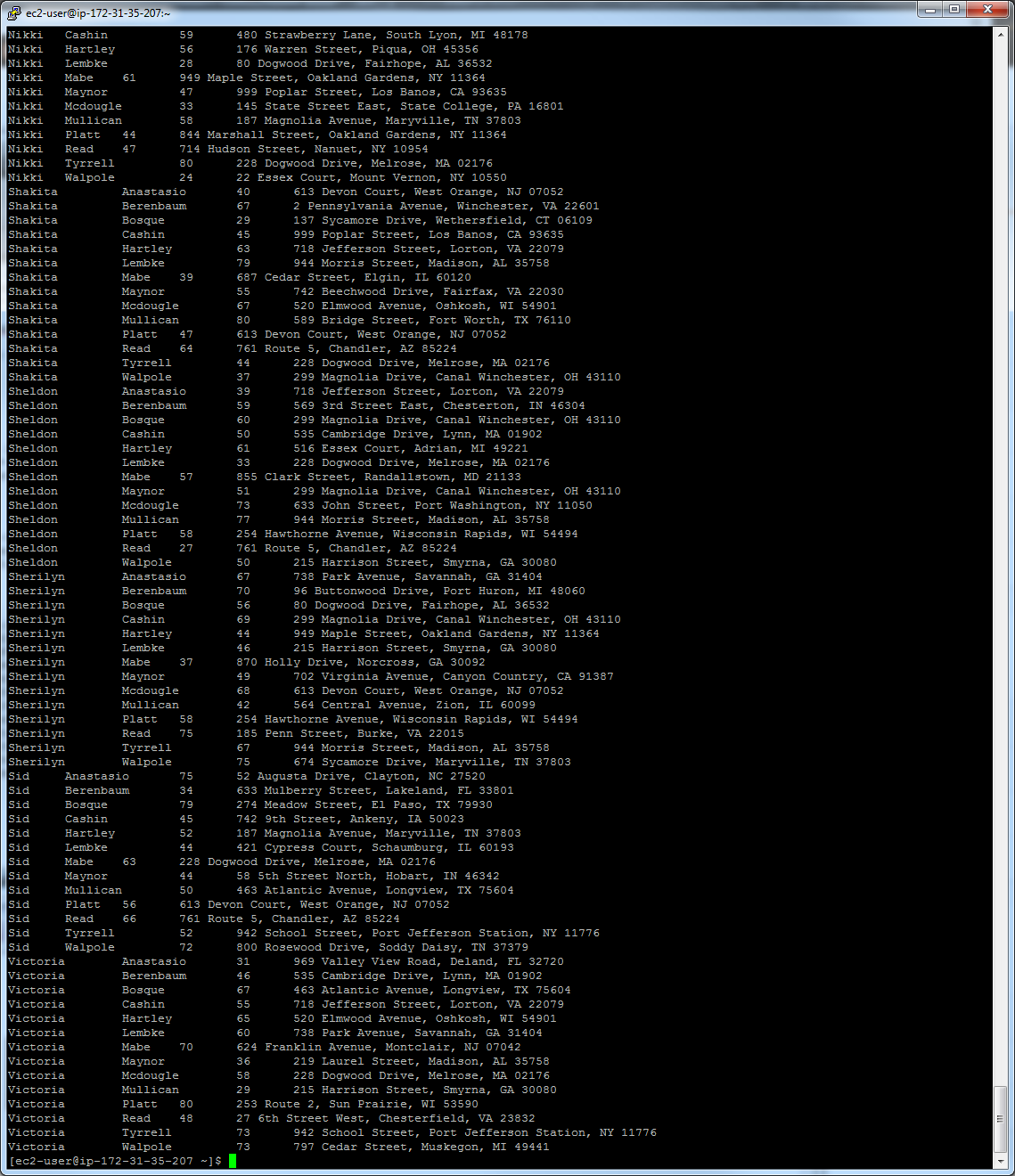
**Directory Contents:**

hadoop fs -ls /data/assignment4\_4



**File Contents (end of file):**

hadoop fs -cat /data/assignment4\_4/part-00000



1. In this section you will practice using HBase and setup Mahout and run the curve-clustering example.
   1. Note that HBase runs on top of HDFS, bypassing MapReduce (so only NameNode and DataNode need to be running). You can use your 4-node cluster or the 1-node cluster to run HBase, but be sure to specify which one you used.

For this exercise I elected to use my 1-node instance.

cd

(Download HBase)

wget http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/hbase-0.90.3.tar.gz

gunzip hbase-0.90.3.tar.gz

tar xvf hbase-0.90.3.tar

cd hbase-0.90.3

(Start HBase service, there is a corresponding stop service and this assumes Hadoop home is set)

bin/start-hbase.sh

(Open the HBase shell – at this point jps should show HMaster)

bin/hbase shell

(Create an employee table and two column families – private and public. Please watch the quotes, if **'** turns into **‘**, the commands will not work)

**create 'employees', {NAME=>** **'private'}, {NAME=>** **'public'}**

**put 'employees', 'ID1', 'private:ssn', '111-222-334'**

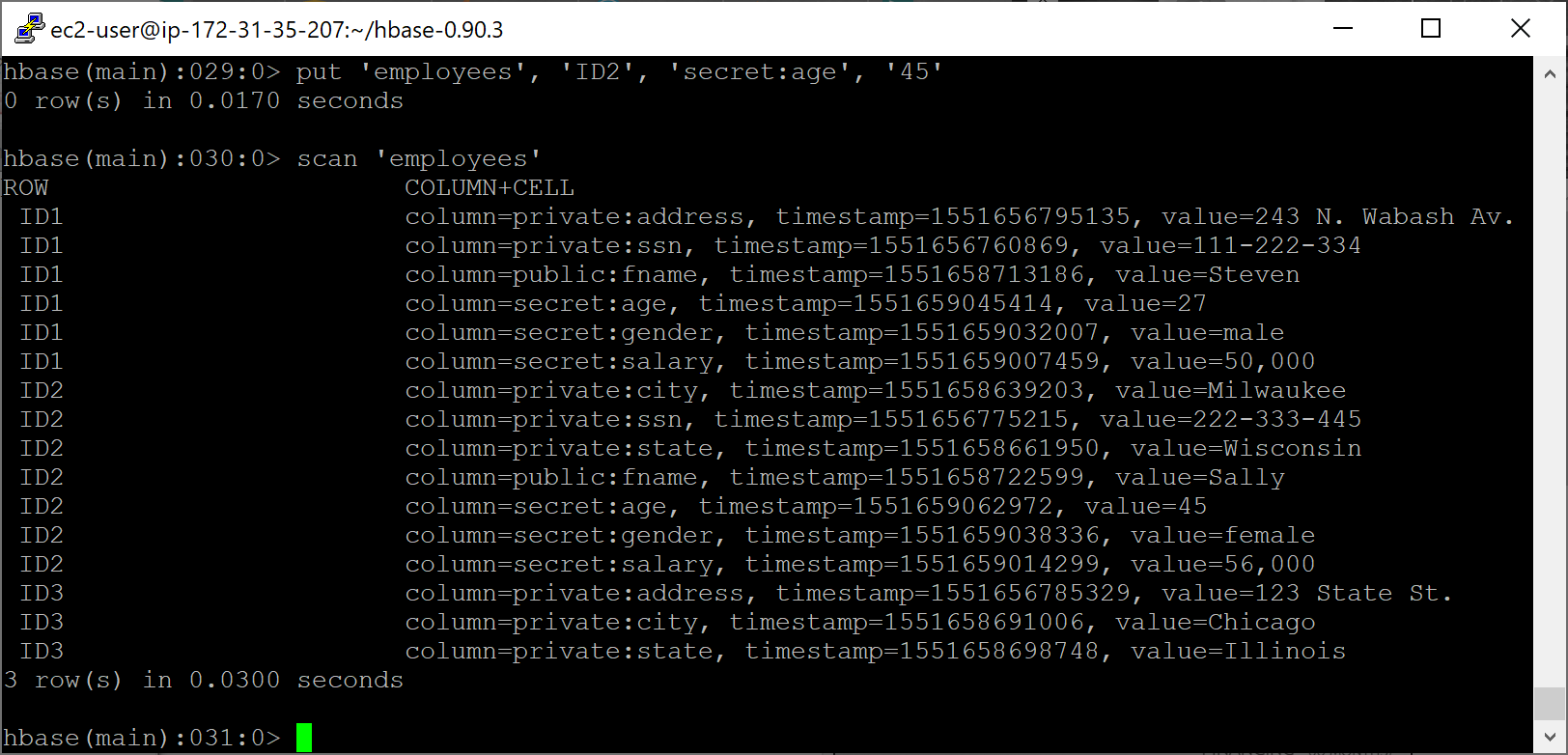
**put 'employees', 'ID2', 'private:ssn', '222-333-445'**

**put 'employees', 'ID3', 'private:address', '123 State St.'**

**put 'employees', 'ID1', 'private:address', '243 N. Wabash Av.'**

**scan 'employees'**

Now that we have filled in a couple of values, add 2 new columns to the private family, 1 new column to the public family and create a brand new family with at least 3 columns. For each of these you should introduce at least 2 values -- so a total of (2+1+3) \* 2 = 12 values inserted. Verify that the table has been filled in properly with scan command and submit a screenshot.



Commands Used:

put 'employees', 'ID2', 'private:city', 'Milwaukee'

put 'employees', 'ID2', 'private:state', 'Wisconsin'

put 'employees', 'ID3', 'private:city', 'Chicago'

put 'employees', 'ID3', 'private:state', 'Illinois'

put 'employees', 'ID1', 'public:fname', 'Steven'

put 'employees', 'ID2', 'public:fname', 'Sally'

disable 'employees'

alter 'employees', {NAME => 'secret'}

enable 'employees'

put 'employees', 'ID1', 'secret:salary', '50,000'

put 'employees', 'ID2', 'secret:salary', '56,000'

put 'employees', 'ID1', 'secret:gender', 'male'

put 'employees', 'ID2', 'secret:gender', 'female'

put 'employees', 'ID1', 'secret:age', '27'

put 'employees', 'ID2', 'secret:age', '45'

* 1. Download and setup Mahout:

cd

(download mahout zip package)

wget http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/apache-mahout-distribution-0.11.2.zip

(Unzip the file)

unzip apache-mahout-distribution-0.11.2.zip

set the environment variables (as always, you can put these commands in ~/.bashrc to automatically set these variables every time you open a new connection, source ~/.bashrc to refresh)

export MAHOUT\_HOME=/home/ec2-user/apache-mahout-distribution-0.11.2

export PATH=/home/ec2-user/apache-mahout-distribution-0.11.2/bin:$PATH

be absolutely sure you set Hadoop home variable (if you haven’t):

Download and prepare synthetic data – it represents a list of 2D curves, represented as a 50-point vector.

Download the synthetic data example:

wget <http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/synthetic_control.data>

(make a testdata directory in HDFS, the example KMeans algorithm assumes the data lives there by default.

hadoop fs -mkdir -p testdatals

(copy the synthetic data over to the testdata directory on HDFS side. You can inspect the contents of the file by running nano synthetic\_control.data – as you can see this is a list of 600 vectors, with individual values separated by a space)

hadoop fs -put synthetic\_control.data testdata/

Please be sure to report the runtime of any command that includes “time”

time mahout org.apache.mahout.clustering.syntheticcontrol.kmeans.Job

real 5m16.566s

user 0m14.818s

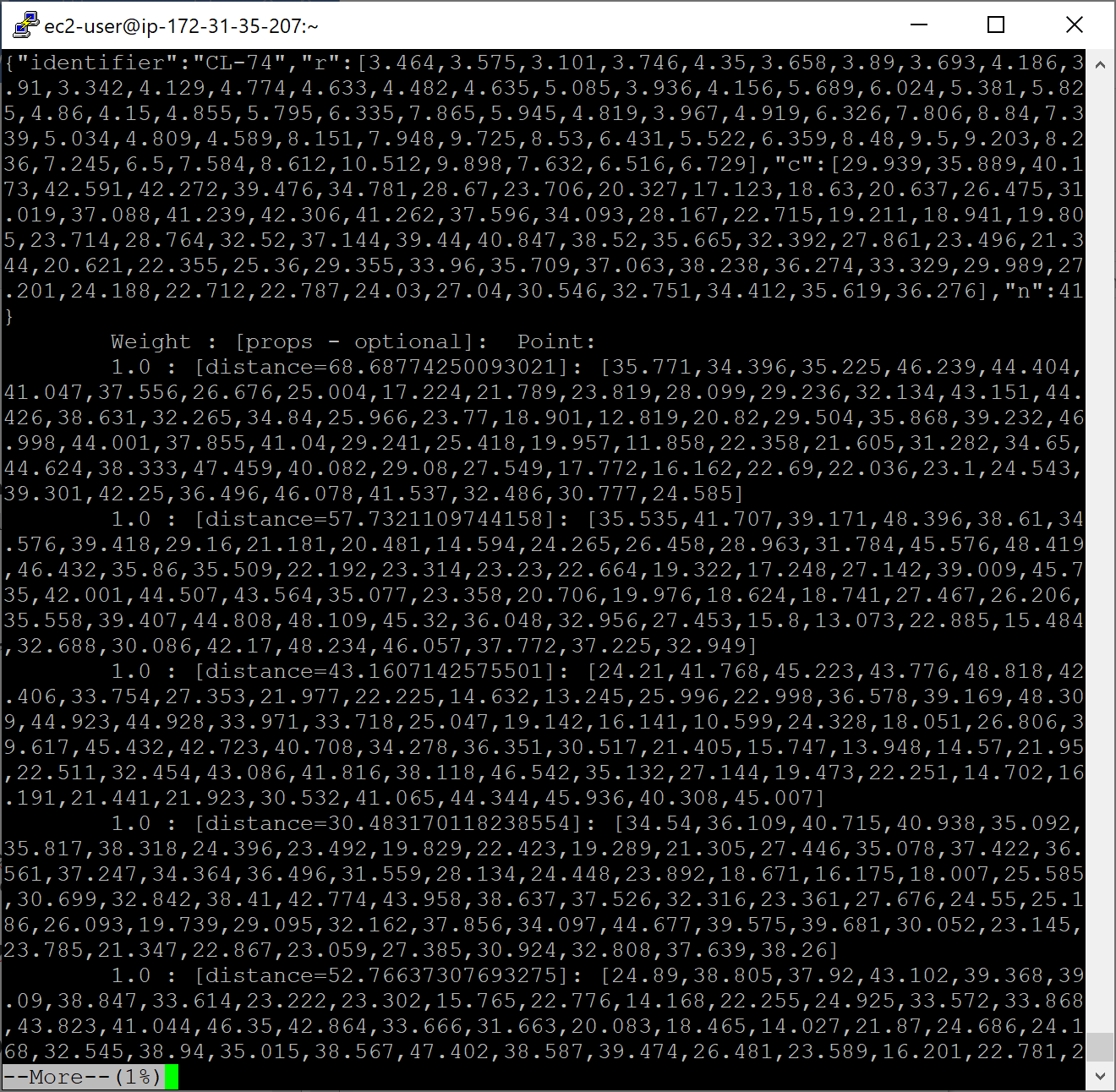
sys 0m3.608s

(clusterdump is a built-in Mahout command that will produce the result of KMeans. Output file is written to clusters-10-final because that is where the output is written after 10 iterations. The center points are placed in a separate file, called clusteredPoints)

mahout clusterdump --input output/clusters-10-final --pointsDir output/clusteredPoints --output clusteranalyze.txt

The file clusteranalyze.txt contains the results of the Kmeans run after 10 iterations.

Submit the screenshot of the first page from clusteranalyze.txt (e.g., from more clusteranalyze.txt)



Submit a single document containing your written answers. Be sure that this document contains your name and “CSC 555 Assignment 4” at the top.

2.a Full Calculation

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 1 | A | 0 | 1 | 1/2 | 1/2 | x | 1/4 | = | 1/2 |
| B | 0 | 0 | 1/2 | 0 | 1/4 | 1/8 |
| Y | 0 | 0 | 0 | 1/2 | 1/4 | 1/8 |
| Z | 1 | 0 | 0 | 0 | 1/4 | 1/4 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 2 | A | 0 | 1 | 1/2 | 1/2 | x | 1/2 | = | 5/16 |
| B | 0 | 0 | 1/2 | 0 | 1/8 | 1/16 |
| Y | 0 | 0 | 0 | 1/2 | 1/8 | 1/8 |
| Z | 1 | 0 | 0 | 0 | 1/4 | 1/2 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 3 | A | 0 | 1 | 1/2 | 1/2 | x | 5/16 | = | 3/8 |
| B | 0 | 0 | 1/2 | 0 | 1/16 | 1/16 |
| Y | 0 | 0 | 0 | 1/2 | 1/8 | 1/4 |
| Z | 1 | 0 | 0 | 0 | 1/2 | 5/16 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 4 | A | 0 | 1 | 1/2 | 1/2 | x | 3/8 | = | 11/32 |
| B | 0 | 0 | 1/2 | 0 | 1/16 | 1/8 |
| Y | 0 | 0 | 0 | 1/2 | 1/4 | 5/32 |
| Z | 1 | 0 | 0 | 0 | 5/16 | 3/8 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 5 | A | 0 | 1 | 1/2 | 1/2 | x | 11/32 | = | 25/64 |
| B | 0 | 0 | 1/2 | 0 | 1/8 | 5/64 |
| Y | 0 | 0 | 0 | 1/2 | 5/32 | 3/16 |
| Z | 1 | 0 | 0 | 0 | 3/8 | 11/32 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 6 | A | 0 | 1 | 1/2 | 1/2 | x | 25/64 | = | 11/32 |
| B | 0 | 0 | 1/2 | 0 | 5/64 | 3/32 |
| Y | 0 | 0 | 0 | 1/2 | 3/16 | 11/64 |
| Z | 1 | 0 | 0 | 0 | 11/32 | 25/64 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 7 | A | 0 | 1 | 1/2 | 1/2 | x | 11/32 | = | 3/8 |
| B | 0 | 0 | 1/2 | 0 | 3/32 | 11/128 |
| Y | 0 | 0 | 0 | 1/2 | 11/64 | 25/128 |
| Z | 1 | 0 | 0 | 0 | 25/64 | 11/32 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 8 | A | 0 | 1 | 1/2 | 1/2 | x | 3/8 | = | 91/256 |
| B | 0 | 0 | 1/2 | 0 | 11/128 | 25/256 |
| Y | 0 | 0 | 0 | 1/2 | 25/128 | 11/64 |
| Z | 1 | 0 | 0 | 0 | 11/32 | 3/8 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 9 | A | 0 | 1 | 1/2 | 1/2 | x | 91/256 | = | 95/256 |
| B | 0 | 0 | 1/2 | 0 | 25/256 | 11/128 |
| Y | 0 | 0 | 0 | 1/2 | 11/64 | 3/16 |
| Z | 1 | 0 | 0 | 0 | 3/8 | 91/256 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 10 | A | 0 | 1 | 1/2 | 1/2 | x | 95/256 | = | 183/512 |
| B | 0 | 0 | 1/2 | 0 | 11/128 | 3/32 |
| Y | 0 | 0 | 0 | 1/2 | 3/16 | 91/512 |
| Z | 1 | 0 | 0 | 0 | 91/256 | 95/256 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 11 | A | 0 | 1 | 1/2 | 1/2 | x | 183/512 | = | 303/823 |
| B | 0 | 0 | 1/2 | 0 | 3/32 | 87/979 |
| Y | 0 | 0 | 0 | 1/2 | 91/512 | 95/512 |
| Z | 1 | 0 | 0 | 0 | 95/256 | 183/512 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 12 | A | 0 | 1 | 1/2 | 1/2 | x | 303/823 | = | 329/913 |
| B | 0 | 0 | 1/2 | 0 | 87/979 | 9/97 |
| Y | 0 | 0 | 0 | 1/2 | 95/512 | 47/263 |
| Z | 1 | 0 | 0 | 0 | 183/512 | 303/823 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 13 | A | 0 | 1 | 1/2 | 1/2 | x | 329/913 | = | 349/953 |
| B | 0 | 0 | 1/2 | 0 | 9/97 | 68/761 |
| Y | 0 | 0 | 0 | 1/2 | 47/263 | 37/201 |
| Z | 1 | 0 | 0 | 0 | 303/823 | 329/913 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 14 | A | 0 | 1 | 1/2 | 1/2 | x | 349/953 | = | 239/661 |
| B | 0 | 0 | 1/2 | 0 | 68/761 | 37/402 |
| Y | 0 | 0 | 0 | 1/2 | 37/201 | 20/111 |
| Z | 1 | 0 | 0 | 0 | 329/913 | 349/953 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 15 | A | 0 | 1 | 1/2 | 1/2 | x | 239/661 | = | 187/512 |
| B | 0 | 0 | 1/2 | 0 | 37/402 | 10/111 |
| Y | 0 | 0 | 0 | 1/2 | 20/111 | 13/71 |
| Z | 1 | 0 | 0 | 0 | 349/953 | 239/661 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 16 | A | 0 | 1 | 1/2 | 1/2 | x | 187/512 | = | 245/676 |
| B | 0 | 0 | 1/2 | 0 | 10/111 | 13/142 |
| Y | 0 | 0 | 0 | 1/2 | 13/71 | 32/177 |
| Z | 1 | 0 | 0 | 0 | 239/661 | 187/512 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 17 | A | 0 | 1 | 1/2 | 1/2 | x | 245/676 | = | 179/491 |
| B | 0 | 0 | 1/2 | 0 | 13/142 | 16/177 |
| Y | 0 | 0 | 0 | 1/2 | 32/177 | 166/909 |
| Z | 1 | 0 | 0 | 0 | 187/512 | 245/676 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 18 | A | 0 | 1 | 1/2 | 1/2 | x | 179/491 | = | 229/631 |
| B | 0 | 0 | 1/2 | 0 | 16/177 | 83/909 |
| Y | 0 | 0 | 0 | 1/2 | 166/909 | 27/149 |
| Z | 1 | 0 | 0 | 0 | 245/676 | 179/491 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 19 | A | 0 | 1 | 1/2 | 1/2 | x | 229/631 | = | 59/162 |
| B | 0 | 0 | 1/2 | 0 | 83/909 | 27/298 |
| Y | 0 | 0 | 0 | 1/2 | 27/149 | 179/982 |
| Z | 1 | 0 | 0 | 0 | 179/491 | 229/631 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 20 | A | 0 | 1 | 1/2 | 1/2 | x | 59/162 | = | 77/212 |
| B | 0 | 0 | 1/2 | 0 | 27/298 | 36/395 |
| Y | 0 | 0 | 0 | 1/2 | 179/982 | 137/755 |
| Z | 1 | 0 | 0 | 0 | 229/631 | 59/162 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 21 | A | 0 | 1 | 1/2 | 1/2 | x | 77/212 | = | 99/272 |
| B | 0 | 0 | 1/2 | 0 | 36/395 | 46/507 |
| Y | 0 | 0 | 0 | 1/2 | 137/755 | 59/324 |
| Z | 1 | 0 | 0 | 0 | 59/162 | 77/212 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 22 | A | 0 | 1 | 1/2 | 1/2 | x | 99/272 | = | 129/355 |
| B | 0 | 0 | 1/2 | 0 | 46/507 | 59/648 |
| Y | 0 | 0 | 0 | 1/2 | 59/324 | 77/424 |
| Z | 1 | 0 | 0 | 0 | 77/212 | 99/272 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 23 | A | 0 | 1 | 1/2 | 1/2 | x | 129/355 | = | 167/459 |
| B | 0 | 0 | 1/2 | 0 | 59/648 | 77/848 |
| Y | 0 | 0 | 0 | 1/2 | 77/424 | 99/544 |
| Z | 1 | 0 | 0 | 0 | 99/272 | 129/355 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 24 | A | 0 | 1 | 1/2 | 1/2 | x | 167/459 | = | 217/597 |
| B | 0 | 0 | 1/2 | 0 | 77/848 | 1/11 |
| Y | 0 | 0 | 0 | 1/2 | 99/544 | 129/710 |
| Z | 1 | 0 | 0 | 0 | 129/355 | 167/459 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 25 | A | 0 | 1 | 1/2 | 1/2 | x | 217/597 | = | 279/767 |
| B | 0 | 0 | 1/2 | 0 | 1/11 | 1/11 |
| Y | 0 | 0 | 0 | 1/2 | 129/710 | 167/918 |
| Z | 1 | 0 | 0 | 0 | 167/459 | 217/597 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 26 | A | 0 | 1 | 1/2 | 1/2 | x | 279/767 | = | 361/993 |
| B | 0 | 0 | 1/2 | 0 | 1/11 | 1/11 |
| Y | 0 | 0 | 0 | 1/2 | 167/918 | 2/11 |
| Z | 1 | 0 | 0 | 0 | 217/597 | 279/767 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 27 | A | 0 | 1 | 1/2 | 1/2 | x | 361/993 | = | 4/11 |
| B | 0 | 0 | 1/2 | 0 | 1/11 | 1/11 |
| Y | 0 | 0 | 0 | 1/2 | 2/11 | 2/11 |
| Z | 1 | 0 | 0 | 0 | 279/767 | 361/993 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 28 | A | 0 | 1 | 1/2 | 1/2 | x | 4/11 | = | 4/11 |
| B | 0 | 0 | 1/2 | 0 | 1/11 | 1/11 |
| Y | 0 | 0 | 0 | 1/2 | 2/11 | 2/11 |
| Z | 1 | 0 | 0 | 0 | 361/993 | 4/11 |
|  |  |  |  |  |  |  |  |  |  |
|  |  | A | B | Y | Z |  | V |  | Rank |
| Step 29 | A | 0 | 1 | 1/2 | 1/2 | x | 4/11 | = | 4/11 |
| B | 0 | 0 | 1/2 | 0 | 1/11 | 1/11 |
| Y | 0 | 0 | 0 | 1/2 | 2/11 | 2/11 |
| Z | 1 | 0 | 0 | 0 | 4/11 | 4/11 |

2.b Full Calculation

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | A | Y | Z |  | V |  | Rank |
| Step 1 | A | 0 | 1 | 1/2 | x | 1/3 | = | 1/2 |
| Y | 0 | 0 | 1/2 | 1/3 | 1/6 |
| Z | 1 | 0 | 0 | 1/3 | 1/3 |
|  |  |  |  |  |  |  |  |  |
|  |  | A | Y | Z |  | V |  | Rank |
| Step 2 | A | 0 | 1 | 1/2 | x | 1/2 | = | 1/3 |
| Y | 0 | 0 | 1/2 | 1/6 | 1/6 |
| Z | 1 | 0 | 0 | 1/3 | 1/2 |
|  |  |  |  |  |  |  |  |  |
|  |  | A | Y | Z |  | V |  | Rank |
| Step 3 | A | 0 | 1 | 1/2 | x | 1/3 | = | 5/12 |
| Y | 0 | 0 | 1/2 | 1/6 | 1/4 |
| Z | 1 | 0 | 0 | 1/2 | 1/3 |
|  |  |  |  |  |  |  |  |  |
|  |  | A | Y | Z |  | V |  | Rank |
| Step 4 | A | 0 | 1 | 1/2 | x | 5/12 | = | 5/12 |
| Y | 0 | 0 | 1/2 | 1/4 | 1/6 |
| Z | 1 | 0 | 0 | 1/3 | 5/12 |
|  |  |  |  |  |  |  |  |  |
|  |  | A | Y | Z |  | V |  | Rank |
| Step 5 | A | 0 | 1 | 1/2 | x | 5/12 | = | 3/8 |
| Y | 0 | 0 | 1/2 | 1/6 | 5/24 |
| Z | 1 | 0 | 0 | 5/12 | 5/12 |
|  |  |  |  |  |  |  |  |  |
|  |  | A | Y | Z |  | V |  | Rank |
| Step 6 | A | 0 | 1 | 1/2 | x | 3/8 | = | 5/12 |
| Y | 0 | 0 | 1/2 | 5/24 | 5/24 |
| Z | 1 | 0 | 0 | 5/12 | 3/8 |
|  |  |  |  |  |  |  |  |  |
|  |  | A | Y | Z |  | V |  | Rank |
| Step 7 | A | 0 | 1 | 1/2 | x | 5/12 | = | 19/48 |
| Y | 0 | 0 | 1/2 | 5/24 | 3/16 |
| Z | 1 | 0 | 0 | 3/8 | 5/12 |
|  |  |  |  |  |  |  |  |  |
|  |  | A | Y | Z |  | V |  | Rank |
| Step 8 | A | 0 | 1 | 1/2 | x | 19/48 | = | 19/48 |
| Y | 0 | 0 | 1/2 | 3/16 | 5/24 |
| Z | 1 | 0 | 0 | 5/12 | 19/48 |
|  |  |  |  |  |  |  |  |  |
|  |  | A | Y | Z |  | V |  | Rank |
| Step 9 | A | 0 | 1 | 1/2 | x | 19/48 | = | 13/32 |
| Y | 0 | 0 | 1/2 | 5/24 | 19/96 |
| Z | 1 | 0 | 0 | 19/48 | 19/48 |
|  |  |  |  |  |  |  |  |  |
|  |  | A | Y | Z |  | V |  | Rank |
| Step 10 | A | 0 | 1 | 1/2 | x | 13/32 | = | 19/48 |
| Y | 0 | 0 | 1/2 | 19/96 | 19/96 |
| Z | 1 | 0 | 0 | 19/48 | 13/32 |
|  |  |  |  |  |  |  |  |  |
|  |  | A | Y | Z |  | V |  | Rank |
| Step 11 | A | 0 | 1 | 1/2 | x | 19/48 | = | 77/192 |
| Y | 0 | 0 | 1/2 | 19/96 | 13/64 |
| Z | 1 | 0 | 0 | 13/32 | 19/48 |
|  |  |  |  |  |  |  |  |  |
|  |  | A | Y | Z |  | V |  | Rank |
| Step 12 | A | 0 | 1 | 1/2 | x | 77/192 | = | 77/192 |
| Y | 0 | 0 | 1/2 | 13/64 | 19/96 |
| Z | 1 | 0 | 0 | 19/48 | 77/192 |
|  |  |  |  |  |  |  |  |  |
|  |  | A | Y | Z |  | V |  | Rank |
| Step 13 | A | 0 | 1 | 1/2 | x | 77/192 | = | 51/128 |
| Y | 0 | 0 | 1/2 | 19/96 | 77/384 |
| Z | 1 | 0 | 0 | 77/192 | 77/192 |
|  |  |  |  |  |  |  |  |  |
|  |  | A | Y | Z |  | V |  | Rank |
| Step 14 | A | 0 | 1 | 1/2 | x | 51/128 | = | 77/192 |
| Y | 0 | 0 | 1/2 | 77/384 | 77/384 |
| Z | 1 | 0 | 0 | 77/192 | 51/128 |
|  |  |  |  |  |  |  |  |  |
|  |  | A | Y | Z |  | V |  | Rank |
| Step 15 | A | 0 | 1 | 1/2 | x | 77/192 | = | 307/768 |
| Y | 0 | 0 | 1/2 | 77/384 | 51/256 |
| Z | 1 | 0 | 0 | 51/128 | 77/192 |
|  |  |  |  |  |  |  |  |  |
|  |  | A | Y | Z |  | V |  | Rank |
| Step 16 | A | 0 | 1 | 1/2 | x | 307/768 | = | 307/768 |
| Y | 0 | 0 | 1/2 | 51/256 | 77/384 |
| Z | 1 | 0 | 0 | 77/192 | 307/768 |
|  |  |  |  |  |  |  |  |  |
|  |  | A | Y | Z |  | V |  | Rank |
| Step 17 | A | 0 | 1 | 1/2 | x | 307/768 | = | 205/512 |
| Y | 0 | 0 | 1/2 | 77/384 | 1/5 |
| Z | 1 | 0 | 0 | 307/768 | 307/768 |
|  |  |  |  |  |  |  |  |  |
|  |  | A | Y | Z |  | V |  | Rank |
| Step 18 | A | 0 | 1 | 1/2 | x | 205/512 | = | 307/768 |
| Y | 0 | 0 | 1/2 | 1/5 | 1/5 |
| Z | 1 | 0 | 0 | 307/768 | 205/512 |
|  |  |  |  |  |  |  |  |  |
|  |  | A | Y | Z |  | V |  | Rank |
| Step 19 | A | 0 | 1 | 1/2 | x | 307/768 | = | 2/5 |
| Y | 0 | 0 | 1/2 | 1/5 | 1/5 |
| Z | 1 | 0 | 0 | 205/512 | 307/768 |
|  |  |  |  |  |  |  |  |  |
|  |  | A | Y | Z |  | V |  | Rank |
| Step 20 | A | 0 | 1 | 1/2 | x | 2/5 | = | 2/5 |
| Y | 0 | 0 | 1/2 | 1/5 | 1/5 |
| Z | 1 | 0 | 0 | 307/768 | 2/5 |
|  |  |  |  |  |  |  |  |  |
|  |  | A | Y | Z |  | V |  | Rank |
| Step 21 | A | 0 | 1 | 1/2 | x | 2/5 | = | 2/5 |
| Y | 0 | 0 | 1/2 | 1/5 | 1/5 |
| Z | 1 | 0 | 0 | 2/5 | 2/5 |