Azat Dovgeldiyev

CSC 555

Assignment 5

Problem 1.

a) Highly structured multi-table data that requires enforcing data constraints.
Relational database.

b) Stock market data ticker with decisions that must be made in real time.
Streaming engine.

c) LinkedIn type data with interconnected nodes where much of the information resides in the links between nodes.
Graph database.

d) An image storage system that allows lookup images by file name.
Key-Value stores

e) A collection of JSON objects (e.g., tweets).
Document-oriented store.

f) Data that is stored in large sparse tables that are continuously growing (new rows/columns).
Column-oriented store

Problem 2.

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

M =

0	1	1/3	1	**A**
0	0	1/3	0	**B**
1	0	0	0	**Y**
0	0	1/3	0	**Z**
A	**B**	**Y**	**Z**	

v =

¼
¼
¼
¼

5 steps, M^5 *v =>

Step 1:

$$\left(0*\frac{1}{4}\right)+\left(1*\frac{1}{4}\right)+\left(\frac{1}{3}*\frac{1}{4}\right)+\left(1*\frac{1}{4}\right)=\frac{7}{12}$$

$$\left(0*\frac{1}{4}\right)+\left(0*\frac{1}{4}\right)+\left(\frac{1}{3}*\frac{1}{4}\right)+\left(0*\frac{1}{4}\right)=\frac{1}{12}$$

$$\left(1*\frac{1}{4}\right)+\left(0*\frac{1}{4}\right)+\left(0*\frac{1}{4}\right)+\left(0*\frac{1}{4}\right)=\frac{1}{4}$$

$$\left(0*\frac{1}{4}\right)+\left(0*\frac{1}{4}\right)+\left(\frac{1}{3}*\frac{1}{4}\right)+\left(0*\frac{1}{4}\right)=\frac{1}{12}$$

Step 2:

$$\left(0*\frac{7}{12}\right)+\left(1*\frac{1}{12}\right)+\left(\frac{1}{3}*\frac{1}{4}\right)+\left(1*\frac{1}{12}\right)=\frac{1}{4}$$

$$\left(0*\frac{7}{12}\right)+\left(0*\frac{1}{12}\right)+\left(\frac{1}{3}*\frac{1}{4}\right)+\left(0*\frac{1}{12}\right)=\frac{1}{12}$$

$$\left(1*\frac{7}{12}\right)+\left(0*\frac{1}{12}\right)+\left(0*\frac{1}{4}\right)+\left(0*\frac{1}{12}\right)=\frac{7}{12}$$

$$\left(0*\frac{7}{12}\right)+\left(0*\frac{1}{12}\right)+\left(\frac{1}{3}*\frac{1}{4}\right)+\left(0*\frac{1}{12}\right)=\frac{1}{12}$$

Step 3:

$$\left(0*\frac{1}{4}\right)+\left(1*\frac{1}{12}\right)+\left(\frac{1}{3}*\frac{7}{12}\right)+\left(1*\frac{1}{12}\right)=\frac{13}{36}$$

$$\left(0*\frac{1}{4}\right)+\left(0*\frac{1}{12}\right)+\left(\frac{1}{3}*\frac{7}{12}\right)+\left(0*\frac{1}{12}\right)=\frac{7}{36}$$

$$\left(1*\frac{1}{4}\right)+\left(0*\frac{1}{12}\right)+\left(0*\frac{7}{12}\right)+\left(0*\frac{1}{12}\right)=\frac{1}{4}$$

$$\left(0*\frac{1}{4}\right)+\left(1*\frac{1}{12}\right)+\left(\frac{1}{3}*\frac{7}{12}\right)+\left(1*\frac{1}{12}\right)=\frac{7}{36}$$

Step 4:

$$\left(0*\frac{13}{36}\right)+\left(1*\frac{7}{36}\right)+\left(\frac{1}{3}*\frac{1}{4}\right)+\left(1*\frac{7}{36}\right)=\frac{17}{36}$$

$$\left(0*\frac{13}{36}\right)+\left(0*\frac{7}{36}\right)+\left(\frac{1}{3}*\frac{1}{4}\right)+\left(0*\frac{7}{36}\right)=\frac{1}{12}$$

$$\left(1*\frac{13}{36}\right)+\left(0*\frac{7}{36}\right)+\left(0*\frac{1}{4}\right)+\left(0*\frac{7}{36}\right)=\frac{13}{36}$$

$$\left(0*\frac{13}{36}\right)+\left(0*\frac{7}{36}\right)+\left(\frac{1}{3}*\frac{1}{4}\right)+\left(0*\frac{7}{36}\right)=\frac{1}{12}$$

Step 5:

$$\left(0*\frac{17}{36}\right)+\left(1*\frac{1}{12}\right)+\left(\frac{1}{3}*\frac{13}{36}\right)+\left(1*\frac{1}{12}\right)=\frac{31}{108}$$

$$\left(0*\frac{17}{36}\right)+\left(0*\frac{1}{12}\right)+\left(\frac{1}{3}*\frac{13}{36}\right)+\left(0*\frac{1}{12}\right)=\frac{13}{108}$$

$$\left(1 * \frac{17}{36}\right) + \left(0 * \frac{1}{12}\right) + \left(0 * \frac{13}{36}\right) + \left(0 * \frac{1}{12}\right) = \frac{17}{36}$$

$$\left(0 * \frac{17}{36}\right) + \left(0 * \frac{1}{12}\right) + \left(\frac{1}{3} * \frac{13}{36}\right) + \left(0 * \frac{1}{12}\right) = \frac{13}{108}$$

Rank :

31/108
13/108
17/36
13/108

b) Now consider a graph with dead-end nodes Q and P:
What is the page rank of Q?
Since p and q are the dead ends, we can drop them:
M = v =

0	1	½	**A**
0	0	½	**Y**
1	0	0	**Z**
A	**Y**	**Z**	

1/3
1/3
1/3

After calculating 5 steps:
V =

3/8
5/24
5/12

Pagerank of Q =

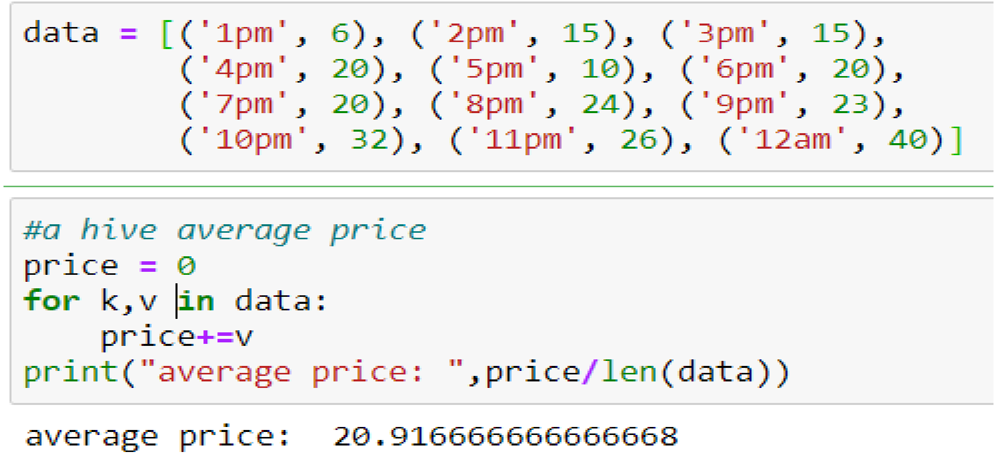
$$\left(0 * \frac{3}{8}\right) + \left(\frac{5}{24} * \frac{1}{2}\right) + \left(\frac{5}{12} * \frac{1}{3}\right) = \frac{35}{144} = 0.243$$

What is the page rank of P?
Now, we can compute the PageRank for P. The node has only one predecessor, Q, and Q has only one successor. Thus, the PageRank of P is the same as that of Q.
1* 35/144 = 0.243

c) If we eliminate all the dead ends, remaining root node has only link to itself and rank 1. First dead end has only the first node as its predecessor, and root node has two successors, so the contribution will be 1/2 to the first dead end, and rank for the first node is 1/2 (as 1*1/2), therefore subsequent dead-end nodes have ranks 1/2.

Problem 3.

a) What will the Hive query "compute average price" return? (yes, this question is as obvious as it seems, asked for comparison with part-b and part-c)

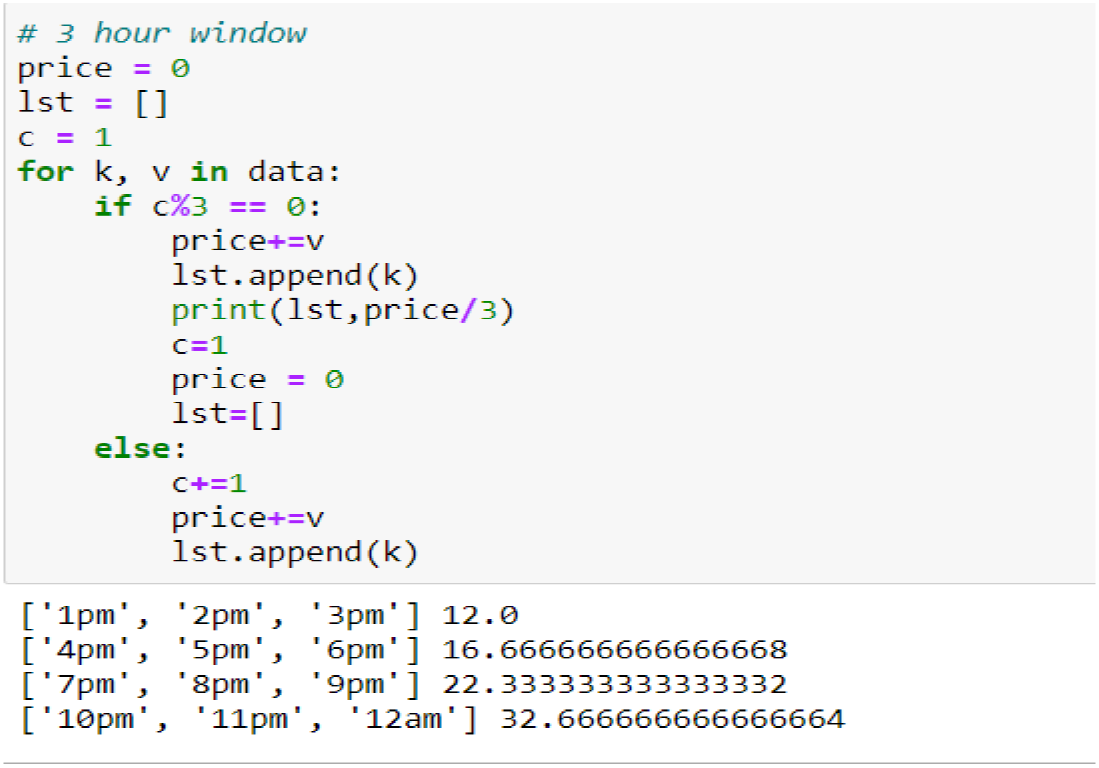


```
data = [('1pm', 6), ('2pm', 15), ('3pm', 15),
        ('4pm', 20), ('5pm', 10), ('6pm', 20),
        ('7pm', 20), ('8pm', 24), ('9pm', 23),
        ('10pm', 32), ('11pm', 26), ('12am', 40)]
```

```
#a hive average price
price = 0
for k,v in data:
    price+=v
print("average price: ",price/len(data))
```

```
average price:  20.916666666666668
```

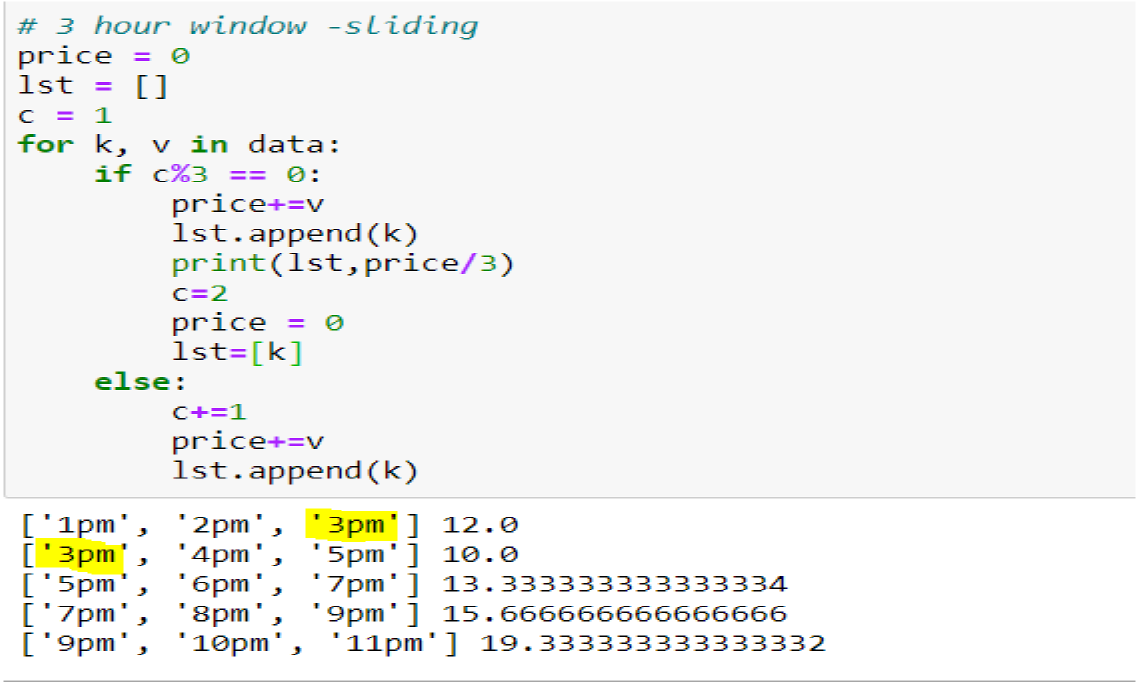
b) What will a Storm streaming query "compute average price per each 3 hour window" return? (tumbling, i.e., non-overlapping window of tuples). For example, the first window would 1pm-4pm. Second window would be 4pm-7pm. If you are wondering about overlap, I would recommend defaulting to (1pm-4pm] (4pm-7pm].



```
# 3 hour window
price = 0
lst = []
c = 1
for k, v in data:
    if c%3 == 0:
        price+=v
        lst.append(k)
        print(lst,price/3)
        c=1
        price = 0
        lst=[]
    else:
        c+=1
        price+=v
        lst.append(k)
```

```
['1pm', '2pm', '3pm'] 12.0
['4pm', '5pm', '6pm'] 16.666666666666668
['7pm', '8pm', '9pm'] 22.333333333333332
['10pm', '11pm', '12am'] 32.666666666666664
```

c) What will a Storm query “compute average price per each 3 hour window” return? (sliding, i.e. overlapping window of tuples, moving the window forward 2 hours each time). First window is 1pm-4pm, second window is 3pm-6pm



```
# 3 hour window -sliding
price = 0
lst = []
c = 1
for k, v in data:
    if c%3 == 0:
        price+=v
        lst.append(k)
        print(lst,price/3)
        c=2
        price = 0
        lst=[k]
    else:
        c+=1
        price+=v
        lst.append(k)
```

```
['1pm', '2pm', '3pm'] 12.0
['3pm', '4pm', '5pm'] 10.0
['5pm', '6pm', '7pm'] 13.333333333333334
['7pm', '8pm', '9pm'] 15.666666666666666
['9pm', '10pm', '11pm'] 19.333333333333332
```

Problem 4.

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:

find . -name “Hadoop-streaming-2.6.4.jar” -print

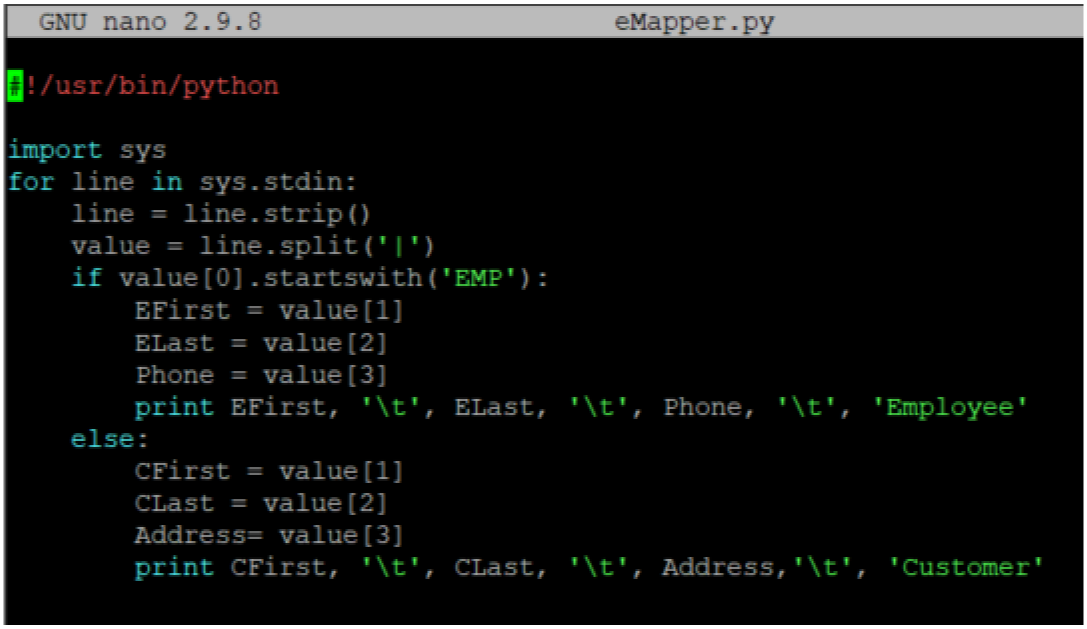
cp .

```
SELECT EFirst, ELast, Phone, Address
FROM Employee, Customer
WHERE EFirst = CFirst AND ELast = Clast
```

eMapper.py

```
#!/usr/bin/python

import sys
for line in sys.stdin:
  line = line.strip()
  value = line.split('|')
  if value[0].startswith('EMP'):
    EFirst = value[1]
    ELast = value[2]
```

```
        Phone = value[3]
        print EFirst, '\t', ELast, '\t', Phone, '\t', 'Employee'
    else:
        CFirst = value[1]
        CLast = value[2]
        Address= value[3]
        print CFirst, '\t', CLast, '\t', Address,'\t', 'Customer'
```

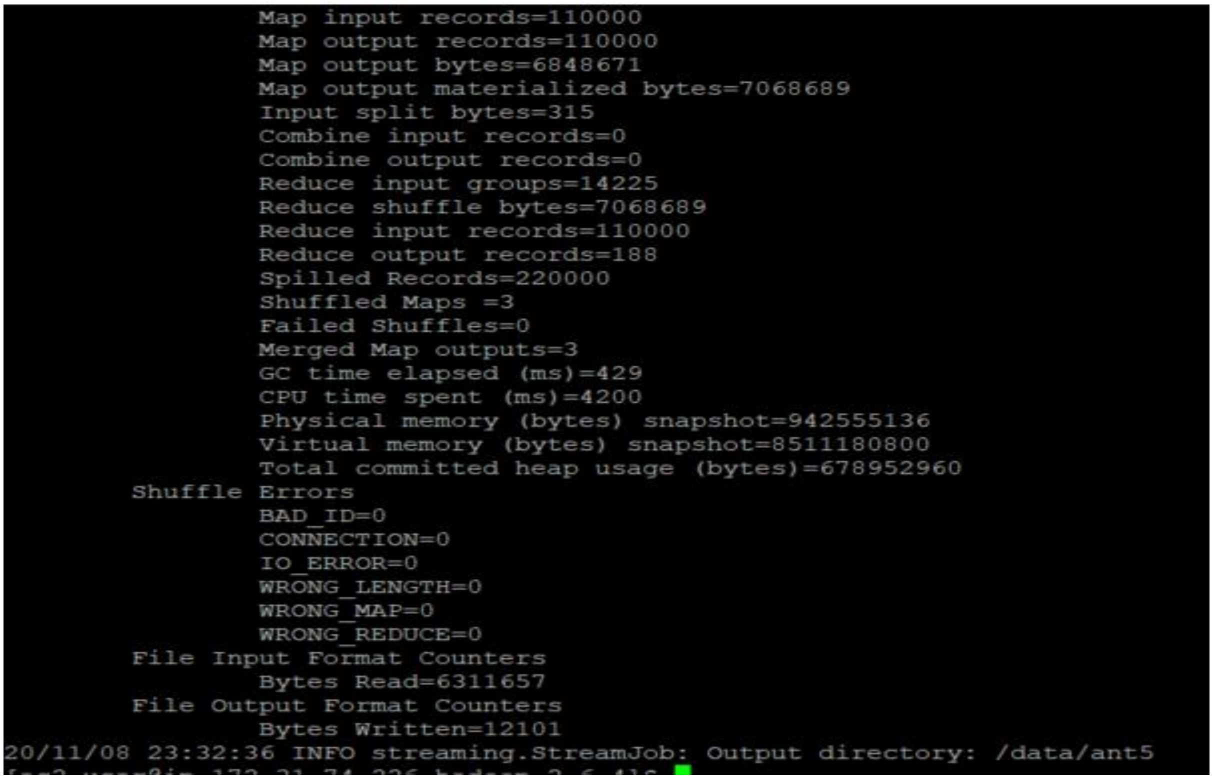


```
  GNU nano 2.9.8                              eMapper.py

#!/usr/bin/python

import sys
for line in sys.stdin:
    line = line.strip()
    value = line.split('|')
    if value[0].startswith('EMP'):
        EFirst = value[1]
        ELast = value[2]
        Phone = value[3]
        print EFirst, '\t', ELast, '\t', Phone, '\t', 'Employee'
    else:
        CFirst = value[1]
        CLast = value[2]
        Address= value[3]
        print CFirst, '\t', CLast, '\t', Address,'\t', 'Customer'
```

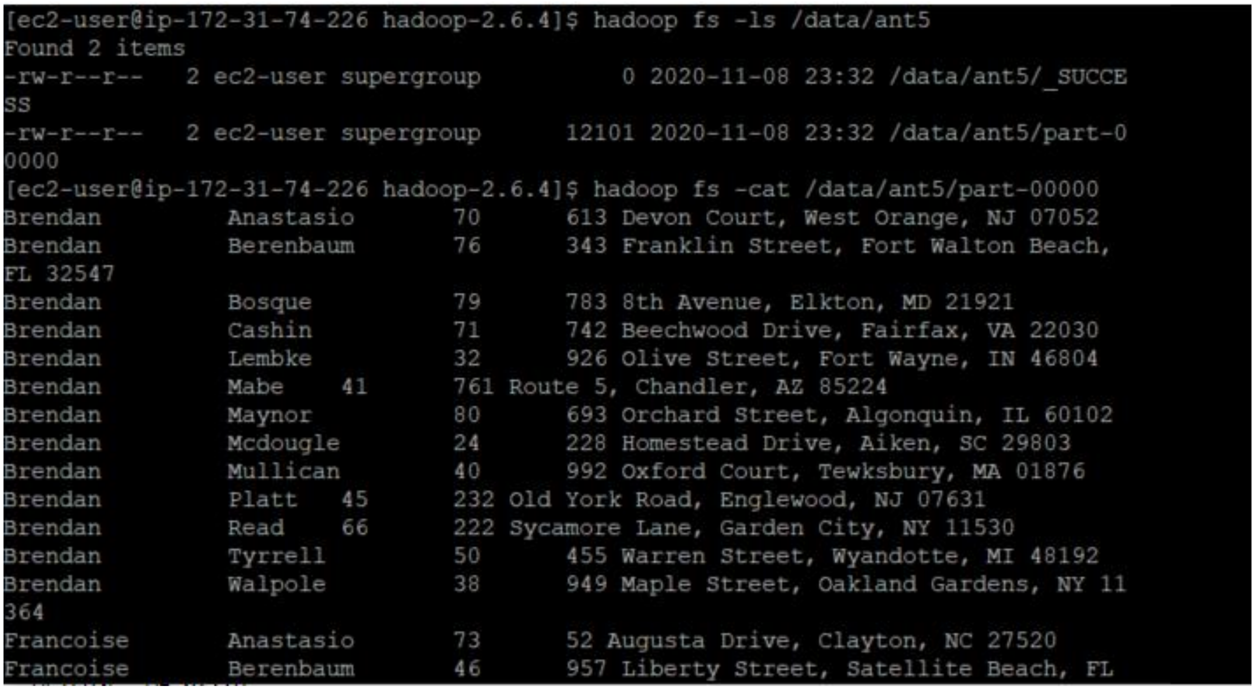
eReducer.py

```
#!/usr/bin/python
import sys

cKey = None
EFirst = None
ELast = None
Phone = None
Address = None

for line in sys.stdin:
    line = line.strip()
    value = line.split('\t')
    k = value[0] + ' ' + value[1]
    v = '\t'.join(value[2:])

    if cKey ==k:
        if v.endswith('Employee'):
            EFirst = value[0]
            ELast = value[1]
            Phone = value[2]
        if v.endswith('Customer'):
            Address = value[2]
    else:
        if cKey:
            lp = len(Phone)
            la = len(Address)
            if (lp * la > 0):
                print EFirst, '\t', ELast, '\t', Phone, '\t', Address
        EFirst = ''
        ELast = ''
        Phone = ''
        Address = ''
        cKey = k
        if v.endswith('Employee'):
            EFirst = value[0]
            ELast = value[1]
            Phone = value[2]
            Address = ''
        elif v.endswith('Customer'):
            EFirst = ''
            ELast = ''
            Phone = ''
            Address = value[2]
            lp = len(Phone)
            la = len(Address)
            if (lp * la > 0):
                print EFirst, '\t', ELast, '\t', Phone, '\t', Address
lp = len(Phone)
la = len(Address)
if (lp * la > 0):
    print EFirst, '\t', ELast, '\t', Phone, '\t', Address
```

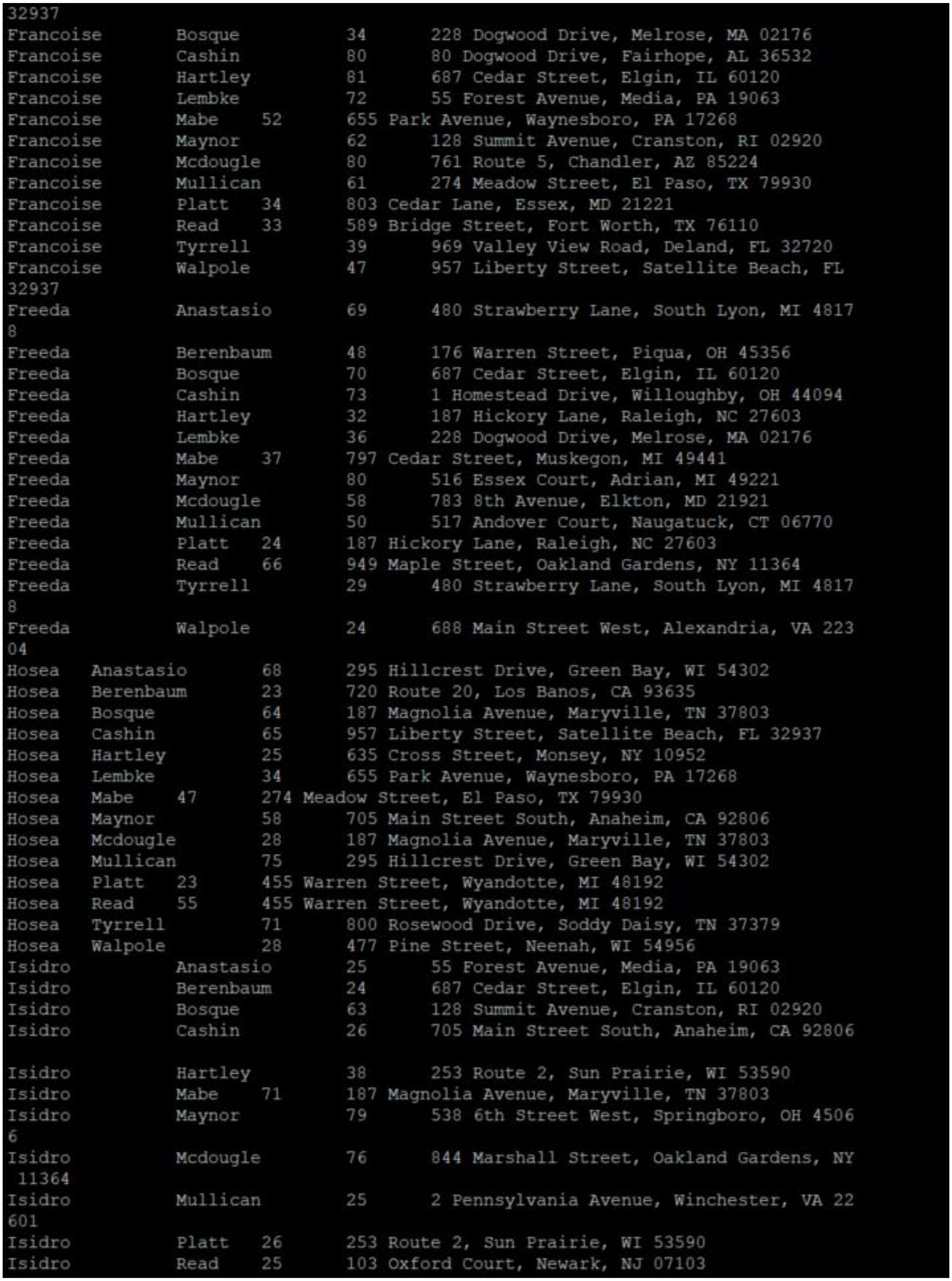
hadoop jar hadoop-streaming-2.6.4.jar-D stream.num.map.output.key.fields=2 -input /user/ec2-user/ecust -output /data/ant5 -mapper eMapper.py -reducer eReducer.py -file eMapper.py -file eReducer.py



```
                Map input records=110000
                Map output records=110000
                Map output bytes=6848671
                Map output materialized bytes=7068689
                Input split bytes=315
                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)=429
                CPU time spent (ms)=4200
                Physical memory (bytes) snapshot=942555136
                Virtual memory (bytes) snapshot=8511180800
                Total committed heap usage (bytes)=678952960
        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
20/11/08 23:32:36 INFO streaming.StreamJob: Output directory: /data/ant5
[ec2-user@ip-172-31-74-226 hadoop-2.6.4]$
```



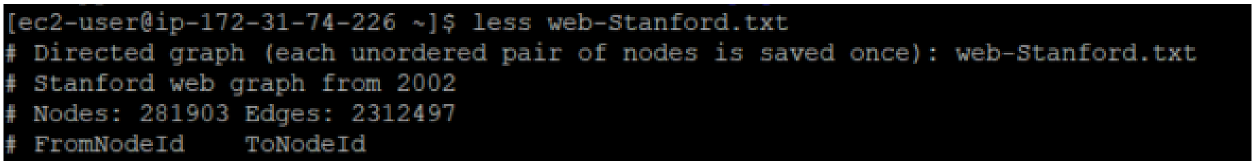
```
[ec2-user@ip-172-31-74-226 hadoop-2.6.4]$ hadoop fs -ls /data/ant5
Found 2 items
-rw-r--r--   2 ec2-user supergroup          0 2020-11-08 23:32 /data/ant5/_SUCCE
SS
-rw-r--r--   2 ec2-user supergroup      12101 2020-11-08 23:32 /data/ant5/part-0
0000
[ec2-user@ip-172-31-74-226 hadoop-2.6.4]$ hadoop fs -cat /data/ant5/part-00000
Brendan         Anastasio       70      613 Devon Court, West Orange, NJ 07052
Brendan         Berenbaum       76      343 Franklin Street, Fort Walton Beach,
FL 32547
Brendan         Bosque          79      783 8th Avenue, Elkton, MD 21921
Brendan         Cashin          71      742 Beechwood Drive, Fairfax, VA 22030
Brendan         Lembke          32      926 Olive Street, Fort Wayne, IN 46804
Brendan         Mabe    41      761 Route 5, Chandler, AZ 85224
Brendan         Maynor          80      693 Orchard Street, Algonquin, IL 60102
Brendan         Mcdougle        24      228 Homestead Drive, Aiken, SC 29803
Brendan         Mullican        40      992 Oxford Court, Tewksbury, MA 01876
Brendan         Platt   45      232 Old York Road, Englewood, NJ 07631
Brendan         Read    66      222 Sycamore Lane, Garden City, NY 11530
Brendan         Tyrrell         50      455 Warren Street, Wyandotte, MI 48192
Brendan         Walpole         38      949 Maple Street, Oakland Gardens, NY 11
364
Francoise       Anastasio       73      52 Augusta Drive, Clayton, NC 27520
Francoise       Berenbaum       46      957 Liberty Street, Satellite Beach, FL
```



```
32937
Francoise       Bosque          34      228 Dogwood Drive, Melrose, MA 02176
Francoise       Cashin          80      80 Dogwood Drive, Fairhope, AL 36532
Francoise       Hartley         81      687 Cedar Street, Elgin, IL 60120
Francoise       Lembke          72      55 Forest Avenue, Media, PA 19063
Francoise       Mabe    52      655 Park Avenue, Waynesboro, PA 17268
Francoise       Maynor          62      128 Summit Avenue, Cranston, RI 02920
Francoise       Mcdougle        80      761 Route 5, Chandler, AZ 85224
Francoise       Mullican        61      274 Meadow Street, El Paso, TX 79930
Francoise       Platt   34      803 Cedar Lane, Essex, MD 21221
Francoise       Read    33      589 Bridge Street, Fort Worth, TX 76110
Francoise       Tyrrell         39      969 Valley View Road, Deland, FL 32720
Francoise       Walpole         47      957 Liberty Street, Satellite Beach, FL
32937
Freeda          Anastasio       69      480 Strawberry Lane, South Lyon, MI 4817
8
Freeda          Berenbaum       48      176 Warren Street, Piqua, OH 45356
Freeda          Bosque          70      687 Cedar Street, Elgin, IL 60120
Freeda          Cashin          73      1 Homestead Drive, Willoughby, OH 44094
Freeda          Hartley         32      187 Hickory Lane, Raleigh, NC 27603
Freeda          Lembke          36      228 Dogwood Drive, Melrose, MA 02176
Freeda          Mabe    37      797 Cedar Street, Muskegon, MI 49441
Freeda          Maynor          80      516 Essex Court, Adrian, MI 49221
Freeda          Mcdougle        58      783 8th Avenue, Elkton, MD 21921
Freeda          Mullican        50      517 Andover Court, Naugatuck, CT 06770
Freeda          Platt   24      187 Hickory Lane, Raleigh, NC 27603
Freeda          Read    66      949 Maple Street, Oakland Gardens, NY 11364
Freeda          Tyrrell         29      480 Strawberry Lane, South Lyon, MI 4817
8
Freeda          Walpole         24      688 Main Street West, Alexandria, VA 223
04
Hosea   Anastasio       68      295 Hillcrest Drive, Green Bay, WI 54302
Hosea   Berenbaum       23      720 Route 20, Los Banos, CA 93635
Hosea   Bosque          64      187 Magnolia Avenue, Maryville, TN 37803
Hosea   Cashin          65      957 Liberty Street, Satellite Beach, FL 32937
Hosea   Hartley         25      635 Cross Street, Monsey, NY 10952
Hosea   Lembke          34      655 Park Avenue, Waynesboro, PA 17268
Hosea   Mabe    47      274 Meadow Street, El Paso, TX 79930
Hosea   Maynor          58      705 Main Street South, Anaheim, CA 92806
Hosea   Mcdougle        28      187 Magnolia Avenue, Maryville, TN 37803
Hosea   Mullican        75      295 Hillcrest Drive, Green Bay, WI 54302
Hosea   Platt   23      455 Warren Street, Wyandotte, MI 48192
Hosea   Read    55      455 Warren Street, Wyandotte, MI 48192
Hosea   Tyrrell         71      800 Rosewood Drive, Soddy Daisy, TN 37379
Hosea   Walpole         28      477 Pine Street, Neenah, WI 54956
Isidro          Anastasio       25      55 Forest Avenue, Media, PA 19063
Isidro          Berenbaum       24      687 Cedar Street, Elgin, IL 60120
Isidro          Bosque          63      128 Summit Avenue, Cranston, RI 02920
Isidro          Cashin          26      705 Main Street South, Anaheim, CA 92806

Isidro          Hartley         38      253 Route 2, Sun Prairie, WI 53590
Isidro          Mabe    71      187 Magnolia Avenue, Maryville, TN 37803
Isidro          Maynor          79      538 6th Street West, Springboro, OH 4506
6
Isidro          Mcdougle        76      844 Marshall Street, Oakland Gardens, NY
 11364
Isidro          Mullican        25      2 Pennsylvania Avenue, Winchester, VA 22
601
Isidro          Platt   26      253 Route 2, Sun Prairie, WI 53590
Isidro          Read    25      103 Oxford Court, Newark, NJ 07103
```

Problem 5.

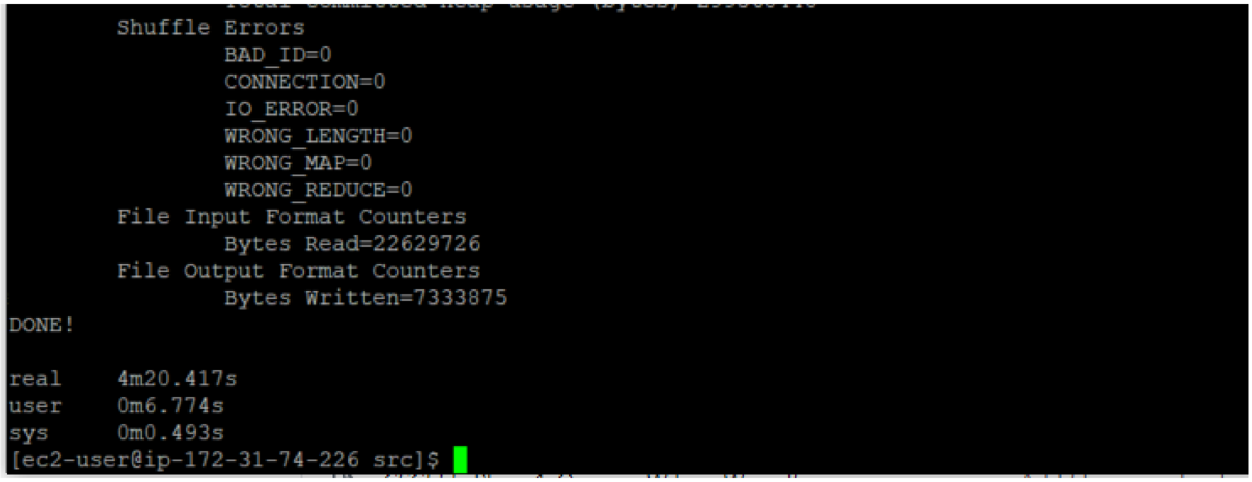
a)



```
[ec2-user@ip-172-31-74-226 ~]$ less web-Stanford.txt
# Directed graph (each unordered pair of nodes is saved once): web-Stanford.txt
# Stanford web graph from 2002
# Nodes: 281903 Edges: 2312497
# FromNodeId    ToNodeId
```

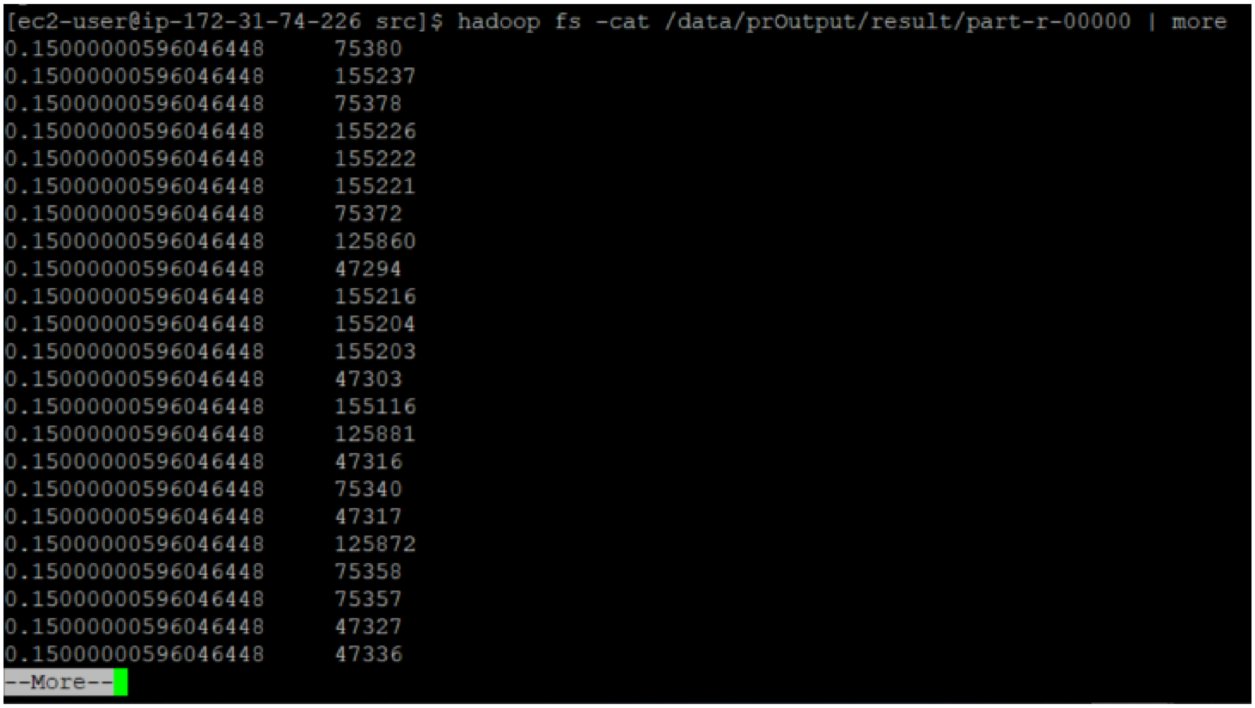
There are 281903 nodes and 2312497 edges the file contains.

b) 4m20. 417s



```
        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=22629726
        File Output Format Counters
                Bytes Written=7333875
DONE!

real    4m20.417s
user    0m6.774s
sys     0m0.493s
[ec2-user@ip-172-31-74-226 src]$
```

c)



```
[ec2-user@ip-172-31-74-226 src]$ hadoop fs -cat /data/prOutput/result/part-r-00000 | more
0.1500000596046448     75380
0.1500000596046448     155237
0.1500000596046448     75378
0.1500000596046448     155226
0.1500000596046448     155222
0.1500000596046448     155221
0.1500000596046448     75372
0.1500000596046448     125860
0.1500000596046448     47294
0.1500000596046448     155216
0.1500000596046448     155204
0.1500000596046448     155203
0.1500000596046448     47303
0.1500000596046448     155116
0.1500000596046448     125881
0.1500000596046448     47316
0.1500000596046448     75340
0.1500000596046448     47317
0.1500000596046448     125872
0.1500000596046448     75358
0.1500000596046448     75357
0.1500000596046448     47327
0.1500000596046448     47336
--More--
```