Robert Kaszubski

CSC 555 – Assignment 5

**1)**

a) Relational Database

b) Streaming Engine

c) Graph Database

d) Key - Value

e) Document – Oriented Store

f) Column – Oriented Store

**2)**

a)

Initial Matrix:

A picture containing text, electronics, keyboard

Description automatically generated

Multiplied by starting Vector:

Text

Description automatically generated

Converges after 50 steps:

Schematic

Description automatically generated with medium confidence

Final Page Ranks:

Text

Description automatically generated

Node A has the highest rank at .4 (4/10)

Node Z has the second highest rank at .3 (3/10)

Node X has the third highest rank at .2 (2/10)

Node B has the lowest rank at .1 (1/10)

Code:

Text

Description automatically generated

b)

When there are dead-end nodes, it will all converge to 0 eventually. So, all nodes will be at 0. Q’s page rank is 0. P’s page rank is 0.

This would be the matrix:

Graphical user interface, text

Description automatically generated

The initial vector:

A picture containing text

Description automatically generated

After 10 steps it gives the following results:

Text, chat or text message

Description automatically generated

After about 100 steps, it gives the following results:

Text

Description automatically generated

Getting close to zero.

After about 2540 steps it converges to 0.

A picture containing schematic

Description automatically generated

What we must do is drop the dead nodes and recompute the page rank for the remaining nodes:

So we drop Q and P

The new matrix for the existing nodes is:

Graphical user interface

Description automatically generated

And the vector is:



This converges to:

Text

Description automatically generated

So the page rank of A and Z is .4

The page rank of X is .2

We can now reintroduce Q.

Page rank of Q = 1/2(.2) + 1/3(.4) = .2333

Its ½ since X points to 2 nodes, and its 1/3 since Z points to 3 nodes.

The Page rank of P is then the same as Q, as it’s the only node linking to P.

Page rank of Q = .2333

Page rank of P = .2333

c)

We have a spider trap linked to series of dead nodes. We can eliminate the dead nodes, and we are left with only the spider trap.

So, the spider trap self-loop node is only linking to itself now. Its page rank is just equal to 1 then. We can’t really teleport to a different node as it’s the only one.

So now we can reintroduce the first node that was previously linked to it. The page rank would be ½(1) = ½ or .5.

Since each subsequent node is a dead-end node and has only one link (the previous node). The page rank of each dead-end node is also .5

**3)**

a)

The uncompressed string is 16 letters long. Each letter takes up 1 byte of space, so the storage size is 16 bytes

b)

The compressed string would look like: Q5Z4A4N3

The new size is 8 bytes

c)

Each letter is 5 bits. 16 ∙ 5 = 80 bits

Each byte is 8 bits, so 80/8 = 10 bytes

The size is 10 bytes.

d)

Run Length Encoding:

Q2B1Q2Z2U1Z1A3N2

The size is now 16 bytes

Dictionary Compressed:

14 x 5 bits = 70 bits

70 bits / 8 = 8.75 bytes

The size is 70 bits or 8.75 bytes

**4)**

a)

(6+15+16+29+10+20+20+21+23+28+26+30)/12

= $20.33

b)

[1pm-4pm) = (6+15+16)/3 = $12.33

[4pm-7pm) = (29+10+20)/3 = $19.67

[7pm-10pm) = (20+21+23)/3 = $21.33

[10pm -1am) = (28+26+30)/3 = $28

c)

[1pm-4pm) = (6+15+16)/3 = $12.33

[3pm-6pm) = (16+29+10)/3 = $18.33

[5pm-8pm) = (10+20+20)/3 = $16.67

[7pm-10pm) = (20+21+23)/3 = $21.33

[9pm-12am) = (23+28+26)/3 = $25.67

[11pm-2am) = No output as it isn’t a full window. We’d have to wait for data from 1am.

**5)**

hadoop fs -mkdir -p /data/hw5

hadoop fs -put employee.txt customer.txt /data/hw5

Command Line:

hadoop jar hadoop-streaming-2.6.4.jar -input /data/hw5 -mapper HW5Mapper.py -file ../HW5Mapper.py -reducer HW5Reducer.py -file ../HW5Reducer.py -output /data/hw5/result

Code: (also attached to submission)

HW5Mapper.py

Text

Description automatically generated

Key is FirstName\_LastName for both input files.

Value is Extention\_EMP for employee input

Value is Address\_CUS for customer input

The employee table’s first column always started with EMP, so I check for that using .startswith() in the first index. This determines which table is being processed. The code is identical for both tables except I differentiate by adding ‘\_EMP’ or ‘\_CUS’.

HW5Reducer.py

Text

Description automatically generated

The Key is FirstName\_LastName

The Value is Address\_Extension

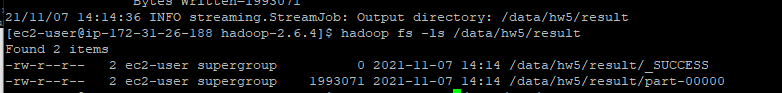
I use the ‘EMP’ or ‘CUS’ to determine which input table each key value pair is from. A join only works if there is a matching key from both tables. If there is, I run a nested for loop to print out all the possible combinations of address and extension as some people have multiple present in the data.

Hadoop Streaming runs successfully

Text

Description automatically generated

Output file:



Sample Output:

Background pattern

Description automatically generated

The Key is FirstName\_LastName

The Value is Address\_Extension

**6)**

a)

The file contains 281903 Nodes, 2312497 Edges.



b)

Text

Description automatically generated

Took 5 minutes and 35 seconds to run.

c)

A picture containing graphical user interface

Description automatically generated