Felipe Guth – 14210231

B. Question:

Suppose that we use MapReduce programming model to code the programming problems above, answer the following questions:

1. Define the input and the output of the Map function(s) for each programming problem. Justify your answer.

Problem I: Matrix Multiplication

A(l,m) x B(m,n)

Map Input:

Values: A(i:l x j:m) and B(i:m,j:n)

The input of the map function are the values correspondent to each pair (i,j) of the matrices A and B.

Map Output:

Keys + values: [(i,k) , A(i,j) , j] and [(i,k) , B(j,k) , j]

The output of the map function generates keys values pairs linking the elements of the lines of A with the columns of B. The columns of B are linked with the lines of A. k varies from 1 to n. The linking combinations of the same keys result in their value computation by the same reducer (i,k).

Problem II: Dissimilarity Matrix

Map Input:

Values A(n x p). The input are the values (i,j) of the matrix that has dimension of (n x p).

Map Output:

Keys + values: [(i,k),j , A(i,j)]

The output of the map function generates key values pairs where each value element of the matrix is linked with the reducer that will calculate the similarity of the elements of two distinct lines. Each value is send to its respective reducers (i,k). j represent the order of each element.

2. Define the input and the output of the Reduce function(s) for each programming problem. Justify your answer.

Problem I: Matrix Multiplication

Reduce Input:

Keys + values: [(i,k) , A(i,j) , j] and [(i,k) , B(j,k) , j]

The input of the reducer is the keys + values generated by the map function. The elements generates by the map jobs, are processed accordingly with the key (i,k). The elements of the same key (i,k) must be multiplied and summed to produce an output of a given reducer.

Reduce Output:

[(i,k), pik]

pik represents the sum of the multiplications of the elements of a correspondent line of A and column of B that are designated with a key (i,k) in the reducer input. So as a result matrix is generated where each key (i,k) correspond to a (line, column) that as a pik value.

Problem II: Dissimilarity Matrix

Reduce Input:

Keys + values: [(i,k),j , A(i,j)]

The input of the reduce function are the respective keys, values generated by the map function. This keys values are taken and used to calculate the difference between elements order by j, that have the same key i,k.

Reduce Output:

[(i,k), dik]

The result of the reduce function are the values of dik for each key. dik is the computation of the dissimilarity between two different objects of dimensions p, that are exhibit in the resulting matrix in the position i,k.

Programming codes

PROBLEM I - Matrix Multiplication

```
#testing with 2x2 * 2x2 matrices on linux - python: cat A.txt B.txt |
./mapper.py 2 2 2 | ./reducer.py
```

#code to read – generate the matrices

```
#!/usr/bin/python

#read and generate file
import sys;
import re;
import random

n = int(raw_input("Enter the number of rows of A:"))
m = int(raw_input("Enter the number of columns of A and rows of B:"))
k = int(raw_input("Enter the number of columns of B:"))


a = [[0 for col in range(m)] for row in range(n)]
b = [[0 for col in range(k)] for row in range(m)]

for i in range(n):
    for j in range(m):
        a[i][j] = raw_input("A enter element %d %d " %(i,j))

    #save to file
    orig_stdout = sys.stdout
    f = file('A.txt', 'w')
    sys.stdout = f

    for i in range(n):
            for j in range(m):
                print("%s %d %d\t%s" %("a",i, j, a[i][j]))

    sys.stdout = orig_stdout
    f.close()

for i in range(m):
    for j in range(k):
        b[i][j] = raw_input("B enter element %d %d " %(i,j))

    #save to file
    orig_stdout = sys.stdout
    f = file('B.txt', 'w')
    sys.stdout = f

    for i in range(m):
            for j in range(k):
                print("%s %d %d\t%s" %("b",i, j, b[i][j]))

    sys.stdout = orig_stdout
    f.close()
```

#--

#MAP code

```
#!/usr/bin/python

import sys;
import re;

#dimensions of mats
l = int(sys.argv[1]);
```

```
m = int(sys.argv[2]);
n = int(sys.argv[3]);

A = [[0 for row in range(m)] for col in range(l)]
B = [[0 for row in range(n)] for col in range(m)]


#import mat
for line in sys.stdin:
    (mat, i, j, v) = re.split("[ \t]+", line.strip());
    i = int(i)
    j = int(j)
    v = int(v)
    if mat == "a":
        A[i][j] = v
    elif mat == "b":
        B[i][j] = v

#produce output map - key value

#A
for i in range(0,l):
    for j in range(0,m):
        for k in range(0,n):
            print("%d %d \t%d %d %s" % (i, k, A[i][j],j,"L"))
            #print("%d %s %s\t%s R" % (c, i, j, v));

#B
for j in range(0,m):
    for k in range(0,n):
        for i in range(0,l):
            print("%d %d \t%d %d %s" % (i, k, B[j][k], j, "R"))
```

#---

#REDUCE CODE

```
#!/usr/bin/python

import sys;
import re;

key = None;

conta = 0;
contb = 0;

import numpy as np


A = []
B = []

for line in sys.stdin:
    (k1, k2, v, j, side) = re.split("[ \t]+", line.strip());

    if side == "L":
        A.append((k1+k2+j,v))
        conta = conta +1;
    elif side == "R":
        B.append((k1+k2+j,v))
        contb = contb +1;
data = np.array(A)
col = 0
A = data[np.argsort(data[:,col])]
data = np.array(B)
col = 0
B = data[np.argsort(data[:,col])]

pik = 0;
auxkey = None
currentkey = None

auxkey = A[0][0]
key = auxkey[:2]

for i in range(len(A)):
    auxkey = A[i][0]
    currentkey = auxkey[:2]

    if key == currentkey:
        pik = pik + int(A[i][1])*int(B[i][1])
    else:
        print(key,pik)
        key = currentkey
        pik = 0 + int(A[i][1])*int(B[i][1])

#print last sum
print(key,pik)
```

--

PROBLEM II – DISISIMILARITY MATRIX

Testing with matrix that contains 3 object and 2 dimensions on linux - python: `cat A.txt | ./mapper.py 3 2 | ./reduce.py`

# Code to read - generate the matrix

```
#!/usr/bin/python

#read and generate file
import sys;
import re;
import random

n = int(raw_input("Enter the number of Objects n:"))
m = int(raw_input("Enter the number dimensions p:"))

a = [[0 for col in range(m)] for row in range(n)]

for i in range(n):
    for j in range(m):
        a[i][j] = raw_input("A enter element %d %d " %(i,j))
    #save to file
    orig_stdout = sys.stdout
    f = file('A.txt', 'w')
    sys.stdout = f

    for i in range(n):
            for j in range(m):
                print("%d %d\t%s" %(i, j, a[i][j]))

    sys.stdout = orig_stdout
    f.close()
```

# MAP

```
#!/usr/bin/python

import sys;
import re;

#dimensions of mats
n = int(sys.argv[1]);
p = int(sys.argv[2]);

A = [[0 for col in range(p)] for row in range(n)]


#import mat
for line in sys.stdin:
    (i, j, v) = re.split("[ \t]+", line.strip());
    i = int(i)
    j = int(j)
    v = int(v)
    A[i][j] = v

#produce output map - key value

for no in range(1,n):
    aux =0
    #while(aux != no):
    for aux in range(0,p):
        if aux != no:
            for ne in range(0,p):
                print("%d %d \t%d" % (no,aux,A[no][ne]))
                #print(no,aux,A[no][ne])
                print("%d %d \t%d" % (no,aux, A[aux][ne]))
        aux = aux + 1;
```

#--

#REDUCE CODE

```
#!/usr/bin/python

import sys;
import re;
import math;

key = None;
dik = 0.0;
auxkey = None
currentkey = None
cont = 0;
aux = 0;
#values are organized in order

#take first key value
for line in sys.stdin:
    (k1, k2, v) = re.split("[ \t]+", line.strip());
    key = k1+k2
    aux = int(v)
    cont = 1;
    break

for line in sys.stdin:
    (k1, k2, v) = re.split("[ \t]+", line.strip());

    currentkey = k1+k2

    if key == currentkey:
        if cont == 0:
            aux = int(v)
            cont = 1;
        else:
            dik = dik + ( aux - int(v))*( aux - int(v))
            cont = 0;
    else:
        print(key,  math.sqrt(dik))
        key = currentkey
        aux = int(v)
        cont = 1
        dik = 0


#print last calc
print(key,math.sqrt(dik))



# ----------------------------------------------------------
```