**Ayush Goyal**

**190905522 CSE D 62**

**Distributed Systems Week 5: Lab 5: Map Reduce Programming using Python**

**Mapper:** A block of data is read and processed to produce key-value pairs as intermediate output. The output of mapper is given as input to reducer.

**Reducer:** Receives the key-value pair from multiple mappers. Then, the reducer aggregates those intermediate data tuples (intermediate key-value pair) into a smaller set of tuples or key-value pairs which is the final output.

**1. Write a basic wordcount program.**

**Note**: I have considered the ‘age’ column of the ‘heart\_disease\_data.csv’ file for this problem.

**1mapper.py**

"""mapper.py"""

import sys

import pandas as pd

df = pd.read\_csv('heart\_disease\_data.csv')

for age in df['age']:

print('%s\t%s' %(age, 1))

**1reducer.py:**

"""reducer.py"""

import sys

current\_word = None

current\_count = 0

word = None

for line in sys.stdin:

line = line.strip()

word, count = line.split()

try:

count = int(count)

except ValueError:

continue

if current\_word == word:

current\_count += count

else:

if current\_word:

print(current\_word + '\t' + str(current\_count))

current\_count = count

current\_word = word

print(current\_word + '\t' + str(current\_count))

**Output:**

**command**: python3 1mapper.py | sort | python3 1reducer.py

Text

Description automatically generated

**2. MapReduce program to find frequent words.**

**Note**: I have considered the ‘Country/Region’ column of the ‘covid\_19\_data.csv’ file for this problem.

**2freqmap1.py**

import pandas as pd

df = pd.read\_csv('covid\_19\_data.csv')

for country in df['Country/Region']:

country = country.strip("(),'")

print('%s\t%d' %(country, 1))

**2freqred1.py**

import sys

lastWord = None

sum = 0

for line in sys.stdin:

word, count = line.strip().split('\t', 1)

count = int(count)

if lastWord == None:

lastWord = word

sum = count

continue

if word == lastWord:

sum += count

else:

print('%s\t%d' %(lastWord, sum))

sum = count

lastWord = word

if lastWord == word:

print('%s\t%s' %(lastWord, sum))

**2freqmap2.py**

import sys

for line in sys.stdin:

word, count = line.strip().split('\t', 1)

count = int(count)

print('%d\t%s' %(count, word))

**2freqred2.py**

import sys

mostFreq = []

currentMax = -1

for line in sys.stdin:

count, word = line.strip().split('\t', 1)

count = int(count)

if count > currentMax:

currentMax = count

mostFreq = [word]

elif count == currentMax:

mostFreq.append(word)

for word in mostFreq:

print('%s\t%s' %(word, currentMax))

**Output:**

**command:** python3 2freqmap1.py | sort | python3 2freqred1.py

Text

Description automatically generated

**command**: python3 2freqmap1.py | sort | python3 2freqred1.py | python3 2freqmap2.py | sort | python3 2freqred2.py

Text

Description automatically generated

**3. Map Reduce program to explore the dataset and perform the filtering (typically creating key/value pairs) by mapper and perform the count and summary operation on the instances.**

**Note**: I have considered the ‘cost’ column of the ‘example.txt’ file for this problem.

**3itemmap.py**

import fileinput

for line in fileinput.input():

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

if len(data) == 6:

date, time, location, item, cost, payment = data

print("{0}\t{1}".format(location, cost))

**3itemred.py**

import fileinput

transactions\_count = 0

sales\_total = 0

for line in fileinput.input():

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

if len(data) != 2:

continue

current\_key, current\_value = data

transactions\_count += 1

sales\_total += float(current\_value)

print(transactions\_count, '\t', sales\_total)

**Output:**

**command**: cat example.txt | python3 3itemmap.py | sort

Text

Description automatically generated

**command**: cat example.txt | python3 3itemmap.py | sort | python3 3itemred.py

Text

Description automatically generated

**4. Write a mapper and reducer program for word count by defining separator instead of using ‘\t’.**

**Note**: I have considered the ‘DurationOfCreditInMonths’ column of the ‘German Credit.xlsx’ file for this problem.

**4sepmap.py**

"""A more advanced mapper using python iterators and generators"""

import sys

import pandas as pd

def main(seperator = '\t'):

G = pd.read\_excel('German Credit.xlsx', sheet\_name = 'Sheet1')

for index, row in G.iterrows():

print('%d%s%d' %(row['DurationOfCreditInMonths'], seperator, 1))

if \_\_name\_\_ == '\_\_main\_\_':

main('\t->')

**4sepred.py**

from itertools import groupby

from operator import itemgetter

import sys

def red\_mapper\_output(file, seperator = '\t'):

for line in file:

yield line.rstrip().split(seperator, 1)

def main(seperator = '\t'):

data = red\_mapper\_output(sys.stdin, seperator)

for current\_word, group in groupby(data, itemgetter(0)):

try:

total\_count = sum(int(count) for current\_word, count in group)

print('%s%s%d' %(current\_word, seperator, total\_count))

except ValueError:

pass

if \_\_name\_\_ == '\_\_main\_\_':

main('\t->')

**Output:**

**command**: python3 4sepmap.py

Text

Description automatically generated

**command**: python3 4sepmap.py | sort | python3 4sepred.py

Text

Description automatically generated

**5. Write a map reduce program that returns the cost of the item that is most expensive, for each location in the dataset example.txt.**

**5itemmap\_expensive.py**

import fileinput

for line in fileinput.input():

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

if len(data) == 6:

date, time, location, item, cost, payment = data

print('{0}\t{1}'.format(location, cost))

**5itemred\_expensive.py**

import fileinput

max\_value = 0

old\_key = None

for line in fileinput.input():

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

if len(data) != 2:

continue

current\_key, current\_value = data

if old\_key and old\_key != current\_key:

print(old\_key, '\t', max\_value)

old\_key = current\_key

max\_value = 0

if float(current\_value) > float(max\_value):

max\_value = float(current\_value)

if old\_key != None:

print(old\_key, '\t', max\_value)

**Output:**

**command**: cat example.txt | python3 5itemmap\_expensive.py | sort

Text

Description automatically generated

**command**: cat example.txt | python3 5itemmap\_expensive.py | sort | python3 5itemred\_expensive.py

Text

Description automatically generated

**6. Write a MapReduce program to evaluate value of PI.**

**6mapper\_pi.py**

import sys

def f(x):

return 4.0 / (1.0 + x \* x)

for line in sys.stdin:

words = line.strip().split()

N = int(words[0])

deltaX = 1.0 / N

for i in range(N):

print('1\t%1.10f' %(f(i \* deltaX) \* deltaX))

**6reducer\_pi.py**

import sys

sum = 0

for line in sys.stdin:

line = line.strip()

word, count = line.split('\t', 1)

try:

count = float(count)

except ValueError:

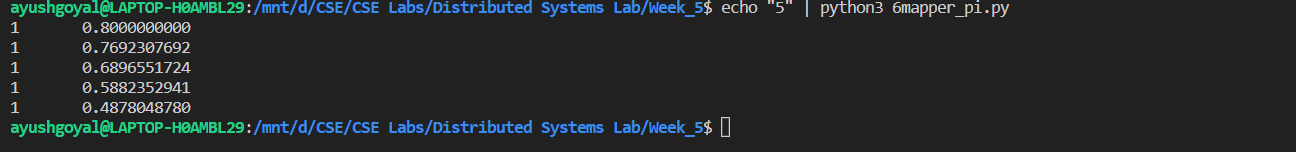
continue

sum += count

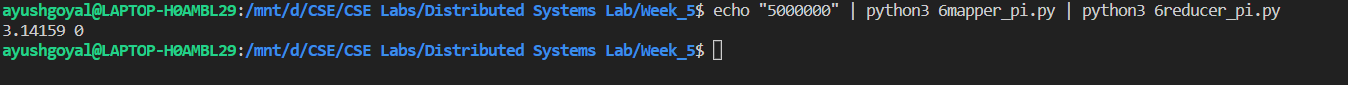
print('%1.5f\t0' %sum)

**Output:**

command: echo "5" | python3 6mapper\_pi.py



command: echo "5000000" | python3 6mapper\_pi.py | python3 6reducer\_pi.py



**7. Write a MapReduce program to count even or odd numbers in randomly generated natural numbers.**

**8mapper1.py**

"""mapper.py"""

import random

random.seed(0)

N = int(input())

for i in range(N):

print(str(random.randrange(1, 10)), '\t', str(1))

**8reducer1.py**

"""reducer.py"""

import sys

lastNumber = 0

count = 0

for line in sys.stdin:

curNumber, curCount = line.strip().split('\t')

curNumber = int(curNumber)

curCount = int(curCount)

if count > 0 and lastNumber != curNumber:

print('%d%s%d' %(lastNumber, '\t', count))

count = 0

lastNumber = curNumber

count += curCount

if count > 0:

print('%d%s%d' %(lastNumber, '\t', count))

**8mapper2.py**

import sys

for line in sys.stdin:

number, count = line.strip().split('\t', 1)

print('%s%s%s' %(count, '\t', number))

**8reducer2.py**

import sys

total = [0] \* 2

for line in sys.stdin:

count, number = line.strip().split('\t', 1)

total[int(number) % 2] += int(count)

print('Even count:\t', str(total[0]))

print('Odd count:\t', str(total[1]))

**Output:**

command: echo "20" | python3 8mapper1.py

Text

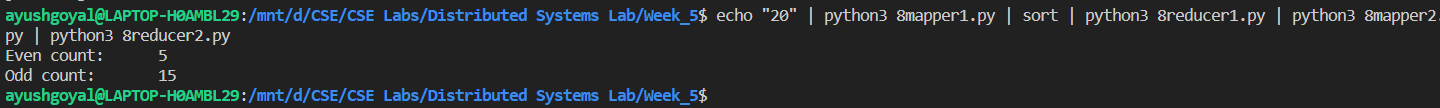
Description automatically generated

command: echo "20" | python3 8mapper1.py | sort | python3 8reducer1.py

Text

Description automatically generated

command: echo "20" | python3 8mapper1.py | sort | python3 8reducer1.py | python3 8mapper2.py | python3 8reducer2.py



**THE END**