EE412 Foundation of Big Data Analytics, Fall 2023

HW3

Name: 권혁태

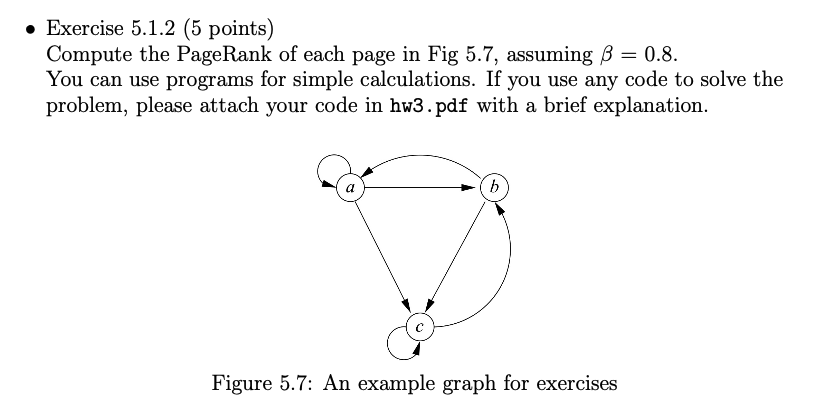
Student ID: 20180036

Discussion Group (People with whom you discussed ideas used in your answers):

On-line or hardcopy documents used as part of your answers:

## Answer to Problem 1

1. Solve following problems



a: 0.25925926

b: 0.30864198

c: 0.43209877

def matrix\_multiply(matrix\_a, matrix\_b):

*# Check if the matrices can be multiplied*

*if* len(matrix\_a[0]) != len(matrix\_b):

*raise* ValueError("Matrices cannot be multiplied. Inner dimensions must match.")

*# Initialize the result matrix with zeros*

result = [[0 *for* \_ *in* range(len(matrix\_b[0]))] *for* \_ *in* range(len(matrix\_a))]

*# Perform matrix multiplication*

*for* i *in* range(len(matrix\_a)):

*for* j *in* range(len(matrix\_b[0])):

*for* k *in* range(len(matrix\_b)):

result[i][j] += matrix\_a[i][k] \* matrix\_b[k][j]

*return* result

def power\_iteration(beta, matrix,random\_walk,num\_iterations):

n = len(matrix)

v = []

*for* i *in* range(n):

v.append([1/n])

*for* i *in* range(num\_iterations):

computed\_matrix = matrix\_multiply(matrix, v)

pageRank = []

*for* i *in* range(n):

pageRank.append([beta \* computed\_matrix[i][0] + (1-beta) \* random\_walk[i][0]])

v = pageRank

*return* v

beta = 0.8

M = [[1/3, 1/2, 0],

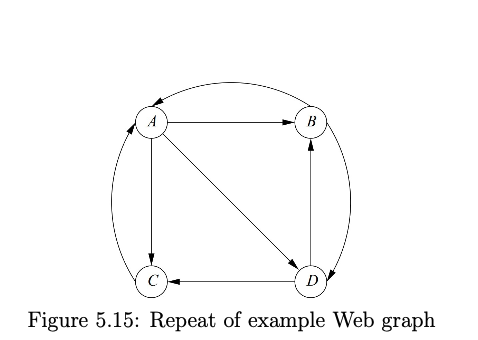
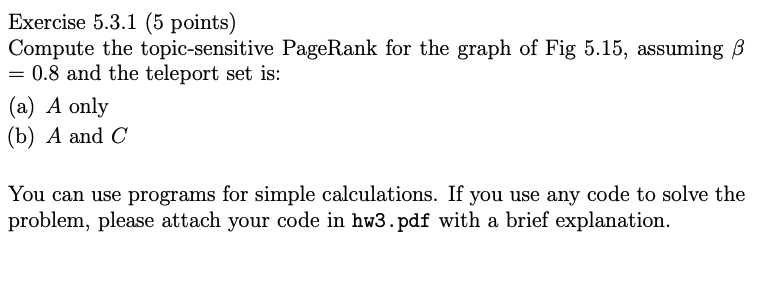
[1/3, 0, 1/2],

[1/3, 1/2, 1/2]]

random = [[1/3],[1/3],[1/3]]

pageRank = power\_iteration(0.8, M, random, 100);

print(pageRank)



1. A only

A: 0.42857143

B: 0.19047619

C: 0.19047619

D: 0.19047619

def matrix\_multiply(matrix\_a, matrix\_b):

*# Check if the matrices can be multiplied*

*if* len(matrix\_a[0]) != len(matrix\_b):

*raise* ValueError("Matrices cannot be multiplied. Inner dimensions must match.")

*# Initialize the result matrix with zeros*

result = [[0 *for* \_ *in* range(len(matrix\_b[0]))] *for* \_ *in* range(len(matrix\_a))]

*# Perform matrix multiplication*

*for* i *in* range(len(matrix\_a)):

*for* j *in* range(len(matrix\_b[0])):

*for* k *in* range(len(matrix\_b)):

result[i][j] += matrix\_a[i][k] \* matrix\_b[k][j]

*return* result

def power\_iteration(beta, matrix,random\_walk,num\_iterations):

n = len(matrix)

v = []

*for* i *in* range(n):

v.append([1/n])

*for* i *in* range(num\_iterations):

computed\_matrix = matrix\_multiply(matrix, v)

pageRank = []

*for* i *in* range(n):

pageRank.append([beta \* computed\_matrix[i][0] + (1-beta) \* random\_walk[i][0]])

v = pageRank

*return* v

beta = 0.8

M = ([[0, 1/2, 1,0 ],

[1/3, 0, 0, 1/2],

[1/3, 0, 0, 1/2],

[1/3, 1/2,0 ,0]]

random = [[1],[0],[0],[0]]

pageRank = power\_iteration(0.8, M, random, 100);

print(pageRank)

1. A and C

A: 0.38571429

B: 0.17142857

C: 0.27142857

D: 0.17142857

def matrix\_multiply(matrix\_a, matrix\_b):

*# Check if the matrices can be multiplied*

*if* len(matrix\_a[0]) != len(matrix\_b):

*raise* ValueError("Matrices cannot be multiplied. Inner dimensions must match.")

*# Initialize the result matrix with zeros*

result = [[0 *for* \_ *in* range(len(matrix\_b[0]))] *for* \_ *in* range(len(matrix\_a))]

*# Perform matrix multiplication*

*for* i *in* range(len(matrix\_a)):

*for* j *in* range(len(matrix\_b[0])):

*for* k *in* range(len(matrix\_b)):

result[i][j] += matrix\_a[i][k] \* matrix\_b[k][j]

*return* result

def power\_iteration(beta, matrix,random\_walk,num\_iterations):

n = len(matrix)

v = []

*for* i *in* range(n):

v.append([1/n])

*for* i *in* range(num\_iterations):

computed\_matrix = matrix\_multiply(matrix, v)

pageRank = []

*for* i *in* range(n):

pageRank.append([beta \* computed\_matrix[i][0] + (1-beta) \* random\_walk[i][0]])

v = pageRank

*return* v

beta = 0.8

M = [[0, 1/2, 1,0 ],

[1/3, 0, 0, 1/2],

[1/3, 0, 0, 1/2],

[1/3, 1/2,0 ,0]]

random = [[1/2],

[0],

[1/2],

[0]]

pageRank = power\_iteration(0.8, M, random, 50);

print(pageRank)

## Answer to Problem 2

텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

텍스트, 친필, 폰트, 문서이(가) 표시된 사진

자동 생성된 설명텍스트, 친필, 폰트, 서예이(가) 표시된 사진

자동 생성된 설명