Lab-1

Name- Tajmim Hossain Purnata

ID- 17101146

Section- 01

## CSE422

**Level-1**

import numpy as np

file =open('C:\\Users\IMADN\Desktop\BFS\BFS.txt','r')

#reading first line for vertex count

line1=file.readline().strip()

vertexcount=int(line1)

#reading 2nd line for edge count

line2=file.readline().strip()

edgecount=int(line2)

#creat adjacent matrix

adjmatrix= np.zeros((vertexcount, vertexcount), dtype='int')

print(adjmatrix)

print(edgecount)

print(vertexcount)

for count in range(edgecount):

line =file.readline().strip()

if not line:

break

vertices=line.split(" ")

u= int(vertices[0])

v= int(vertices[1])

#print(u)

#print(v)

adjmatrix[u][v]+= 1

line3=file.readline().strip()

enemy\_vertex=int(line3)

print(enemy\_vertex)

for row in range(vertexcount):

for col in range(vertexcount):

print(adjmatrix[row][col], end=' ')

print()

color=np.empty((vertexcount), dtype="object")

color[:]="White"

parent=np.empty((vertexcount), dtype="object")

parent[:]=np.NaN

d=np.empty((vertexcount), dtype="int")

d[:]= 999999

#queue using queue model

from queue import Queue

#initializing queue

myqueue= Queue(maxsize= vertexcount)

#bfs function

def bfs(s):

global color, parent, d

color[s] = 'Gray'

parent[s]= np.nan

d[s]= 0

myqueue.put(s)

while not myqueue.empty():

u=myqueue.get() #popping element

#print(u)

for v in range(1, vertexcount):

if adjmatrix[u][v]==1:

if color[v] == 'White':

color[v]='Gray'

d[v]=d[u] +1

parent[v]=u

myqueue.put(v)

if v== enemy\_vertex:

print(d[v])

return

color[u]= 'Black'

#print(u, 'distance: ', d[u])

bfs(0)

**Level-2**

import numpy as np

file = open('C:/Users/IMADN/Desktop/BFS/bfs2.txt', 'r')

#read vertex

line1 = file.readline().strip()

vertexcount=int(line1)

#read edge

line2 = file.readline().strip()

edgecount=int(line2)

#create adjmatrix

adjmatrix=np.zeros((vertexcount, vertexcount), dtype= 'int')

print(adjmatrix)

for count in range(edgecount):

line=file.readline().strip()

if not line:

break

vertices=line.split(" ")

u= int(vertices[0])

v= int(vertices[1])

print(u)

print(v)

adjmatrix[u][v]+=1

adjmatrix[v][u]+=1

#lina

line3=file.readline().strip()

lina=int(line3)

#nora

line4=file.readline().strip()

nora=int(line4)

#lara

line5=file.readline().strip()

lara=int(line5)

print(lina)

print(nora)

print(lara)

#assigning value to matrix

for row in range(vertexcount):

for col in range(vertexcount):

print(adjmatrix[row][col], end=' ')

print()

color=np.empty((vertexcount), dtype="object")

color[:]="White"

parent=np.empty((vertexcount), dtype="object")

parent[:]=np.NaN

d=np.empty((vertexcount), dtype="int")

d[:]= 999999

#queue using queue model

from queue import Queue

#initializing queue

myqueue= Queue(maxsize= vertexcount)

#bfs function

def bfs(s):

global color, parent, d

color[:]='White'

parent[:]=np.NaN

d[:]=999999

while not myqueue.empty():

myqueue.get()

color[s] = 'Gray'

parent[s]= np.nan

d[s]= 0

myqueue.put(s)

while not myqueue.empty():

u=myqueue.get() #popping element

#print(u)

for v in range(1, vertexcount):

if adjmatrix[u][v]==1:

if color[v] == 'White':

color[v]='Gray'

d[v]=d[u] +1

parent[v]=u

myqueue.put(v)

if v== lina:

return d[v]

break

color[u]= 'Black'

noraPosition=(bfs(nora))

laraPosition=(bfs(lara))

if noraPosition < laraPosition:

print("Nora")

if laraPosition < noraPosition:

print("Lara")

if noraPosition == laraPosition:

print("Both")

**Level-3**

import numpy as np

file = open('C:/Users/IMADN/Desktop/BFS/bfs3.txt', 'r')

#read vertex

line1 = file.readline().strip()

vertexcount=int(line1)

#read edge

line2 = file.readline().strip()

edgecount=int(line2)

#create adjmatrix

adjmatrix=np.zeros((vertexcount, vertexcount), dtype= 'int')

print(adjmatrix)

for count in range(edgecount):

line=file.readline().strip()

if not line:

break

vertices=line.split(" ")

u= int(vertices[0])

v= int(vertices[1])

print(u)

print(v)

adjmatrix[u][v]+=1

adjmatrix[v][u]+=1

#lina postion

line3 = file.readline().strip()

lina = int(line3)

#participants

line4 = file.readline().strip()

participant = int(line4)

#each participant position

position = np.zeros(participant, dtype = 'int')

for k in range(0, participant):

line5 = file.readline().strip()

position[k]=int(line5)

for a in range(0, participant):

print(position[a])

color=np.empty((vertexcount), dtype="object")

color[:]="White"

parent=np.empty((vertexcount), dtype="object")

parent[:]=np.NaN

d=np.empty((vertexcount), dtype="int")

d[:]= 999999

#queue using queue model

from queue import Queue

#initializing queue

myqueue= Queue(maxsize= vertexcount)

def bfs(s):

global color, parent, d

color[:]= 'White'

parent[:] = np.NaN

d[:]=999999

while not myqueue.empty():

myqueue.get()

color[s]='Gray'

parent[s]=np.NaN

d[s]=0

myqueue.put(s)

while not myqueue.empty():

u=myqueue.get(s)

for v in range(1, vertexcount):

if adjmatrix[u][v]==1:

if color[v]=='White':

color[v]='Gray'

parent[v]=u

d[v]= d[u]+1

myqueue.put(v)

for count in range(0, participant):

if v == position[count]:

print(d[v])

return

color[u]='Black'