**LEVEL 1:**

from google.colab import files

uploaded = files.upload()

import numpy as np

file = open('lv1.txt','r')

line1 = file.readline()

line2 = file.readline()

vertext\_count = int(line1) +1

x = int(line2)

adj\_matrix = np.zeros((vertext\_count, vertext\_count), dtype = 'int')

print(adj\_matrix)

for i in range(x):

line = file.readline()

vertices = line.split()

u = int(vertices[0])

v = int(vertices[1])

adj\_matrix[u][v] = 1

adj\_matrix[v][u] = 1

print(adj\_matrix)

lina = int(file.readline())

color = np.empty(vertext\_count, dtype = 'object')

color[:] ='WHITE'

parent = np.empty(vertext\_count, dtype = 'object')

parent[:] = np.NaN

d= np.zeros(vertext\_count,dtype = 'int')

from queue import Queue

my\_queue = Queue(maxsize = vertext\_count)

def bfs(g,s):

color[s] = 'Gray'

parent[s] = np.nan

d[s] = 0

my\_queue.put(s)

while not my\_queue.empty():

u = my\_queue.get()

for v in range (1,vertext\_count):

if adj\_matrix[u][v] == 1:

if color[v] == 'WHITE':

color[v] = 'Gray'

d[v] = d[u] + 1

parent[v] = u

my\_queue.put(v)

color[u] = 'BLACK'

#print(str(u) + 'distance' + str(d[u]))

return d

distance = bfs(adj\_martix,1)

print(distance[lina])

**LEVEL 2:**

from google.colab import files

uploaded = files.upload()

import numpy as np

file = open('lv2.txt','r')

line1 = file.readline()

line2 = file.readline()

vertext\_count = int(line1) +1

x = int(line2)

adj\_matrix = np.zeros((vertext\_count, vertext\_count), dtype = 'int')

print(adj\_matrix)

for i in range(x):

line = file.readline()

vertices = line.split()

u = int(vertices[0])

v = int(vertices[1])

adj\_matrix[u][v] = 1

adj\_matrix[v][u] = 1

print(adj\_matrix)

lina = int(file.readline())

nora = int(file.readline())

lara = int(file.readline())

color = np.empty(vertext\_count, dtype = 'object')

color[:] ='WHITE'

parent = np.empty(vertext\_count, dtype = 'object')

parent[:] = np.NaN

d= np.zeros(vertext\_count,dtype = 'int')

from queue import Queue

my\_queue = Queue(maxsize = vertext\_count)

def bfs(g,s):

color[s] = 'Gray'

parent[s] = np.nan

d[s] = 0

my\_queue.put(s)

while not my\_queue.empty():

u = my\_queue.get()

for v in range (1,vertext\_count):

if adj\_matrix[u][v] == 1:

if color[v] == 'WHITE':

color[v] = 'Gray'

d[v] = d[u] + 1

parent[v] = u

my\_queue.put(v)

color[u] = 'BLACK'

#print(str(u) + 'distance' + str(d[u]))

return d

distance = bfs(adj\_matrix,1)

n = bfs(adj\_matrix, nora)[lina]

l =bfs(adj\_matrix,lara)[nora]

if n>l:

print('Nora')

else:

print('Lara')

**LEVEL 3:**

from google.colab import files

uploaded = files.upload()

import numpy as np

file = open('lv3.txt','r')

line1 = file.readline()

line2 = file.readline()

vertext\_count = int(line1) +1

x = int(line2)

adj\_matrix = np.zeros((vertext\_count, vertext\_count), dtype = 'int')

print(adj\_matrix)

for i in range(x):

line = file.readline()

vertices = line.split()

u = int(vertices[0])

v = int(vertices[1])

adj\_matrix[u][v] = 1

adj\_matrix[v][u] = 1

print(adj\_matrix)

lina = int(file.readline())

color = np.empty(vertext\_count, dtype = 'object')

color[:] ='WHITE'

parent = np.empty(vertext\_count, dtype = 'object')

parent[:] = np.NaN

d= np.zeros(vertext\_count,dtype = 'int')

from queue import Queue

my\_queue = Queue(maxsize = vertext\_count)

def bfs(g,s):

color[s] = 'Gray'

parent[s] = np.nan

d[s] = 0

my\_queue.put(s)

while not my\_queue.empty():

u = my\_queue.get()

for v in range (1,vertext\_count):

if adj\_matrix[u][v] == 1:

if color[v] == 'WHITE':

color[v] = 'Gray'

d[v] = d[u] + 1

parent[v] = u

my\_queue.put(v)

color[u] = 'BLACK'

#print(str(u) + 'distance' + str(d[u]))

return d

distance = bfs(adj\_martix,1)

print(distance[k5])