	Bansilal Ramnath Agarwal Charitable Trust's Vishwakarma Institute of Information Technology **Department of Artificial Intelligence and Data Science**

Name: Siddhesh Dilip Khairnar		
Class: SY	Division: B	Roll No: 272028
Semester: IV	Academic Year: 2022-2023	
Subject Name & Code: Advanced Data Structure, ADUA22202		
Title of Assignment: Consider a friends' network on Facebook social website. Model it as a graph to represent each node as a user and a link to represent the friend relationship between them. Store data such as date of birth, number of comments for each User.		

Aim: Consider a friends' network on Facebook social website. Model it as a graph to represent each node as a user and a link to represent the friend relationship. between them. Store data such as date of birth, number of comments for each User:

1. Find who has maximum friends.
2. Find who has posted maximum and minimum comments.

Topic : Assignment 04 (ADS)

Page No.
Date

Name : Siddhesh Khairnar
Division : B
Roll no : 272028

Aim: Consider a friend network on facebook social website. Model it a graph to represent each node as a user and link to represent the friend relation -ship between them. store data such as date of birth, no of comment for each user.

write up:

(i) To model a friend network on facebook as a graph, we can represent each users as node and friend relationship between them as a link or edge.

(ii) we can store data such as date of birth and no. of comment they made as attributes of a node.

Graph data structure : -

- In python there are multiple way to represent graph, some of which are:

i) Adjacency List :-

⇒ An adjacency list is an collection of unordered list used to represent a finite graph. Each list describe that set of neighbour of vertex in graph

ii) Adjacency Matrix :

⇒ An adjacency matrix is 2 D array used to represent a finite graph. The element of matrix represent edges between vertices.

iii) Network x :

⇒ Network x is a python library for working with graph. It provides a variety of data structure and algorithm for graph analysis and manipulation.

Algorithm: -

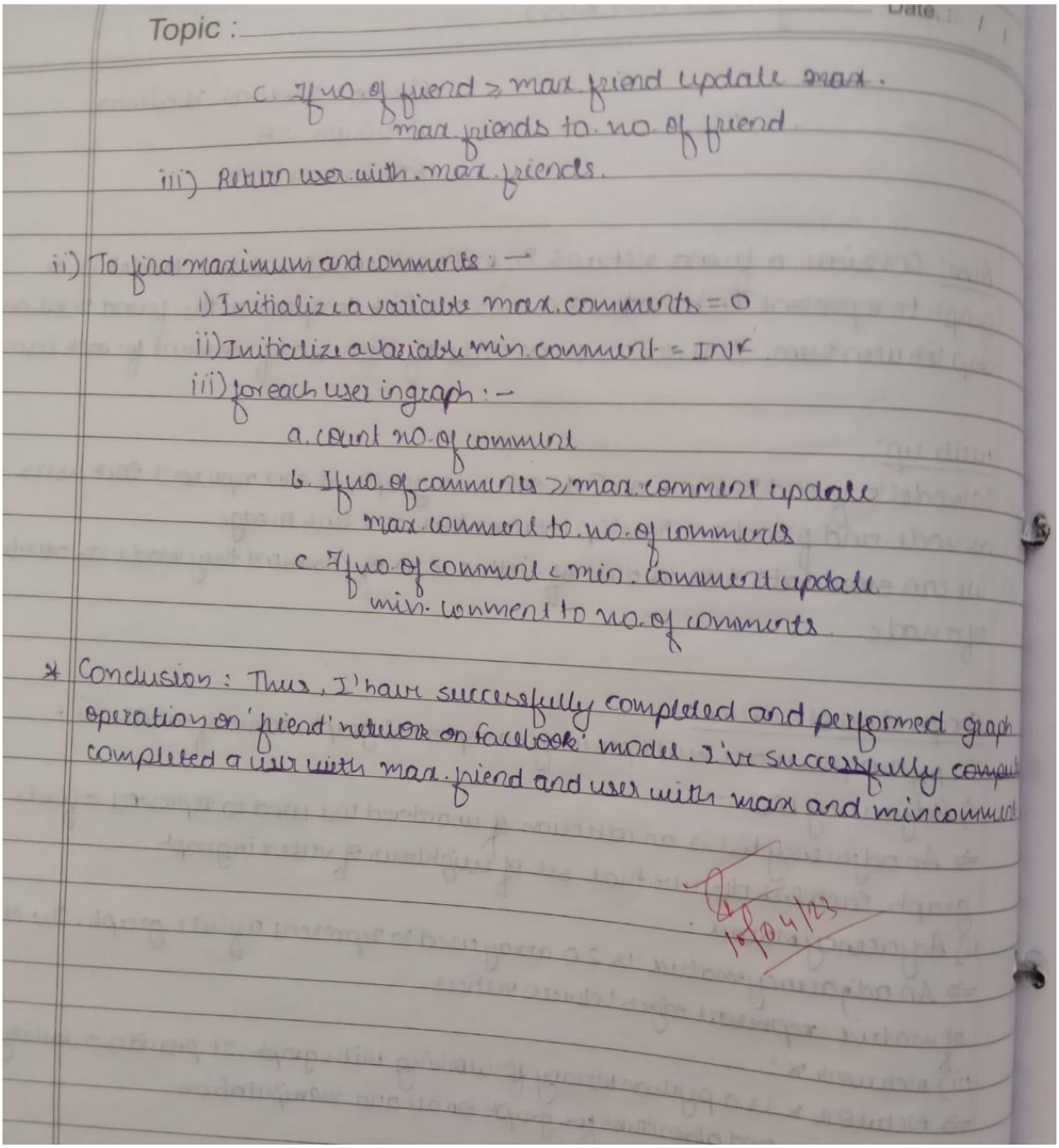
i) To find max friends :

1) Initialize variable max_friend and user_with_friend = null

ii) for each user in graph :

a) Perform traversal of graph

b) Inc count variable as no. of friend.


Topic :

c. If no. of friend > max friend update max. max friends to no. of friend

iii) Return user with max friends.

ii) To find maximum and comments :—

i) Initialize a variable max. comments = 0

ii) Initialize a variable min. comment = INF

iii) for each user in graph :—

a. count no. of comment

b. If no. of comments > max. comment update max. comment to no. of comments

c. If no. of comment < min. comment update min. comment to no. of comments

* Conclusion : Thus, I'have successfully completed and performed graph operation on 'friend' network on facebook' model. I've successfully compu completed a user with max. friend and user with max and min comment

16/04/23

Experiment:

```
import networkx as nx

# Create an empty graph
G = nx.Graph()
```

```
# Add nodes to the graph
G.add_node('User1', dob='2000-01-01', num_comments=100)
G.add_node('User2', dob='2001-03-15', num_comments=90)
G.add_node('User3', dob='2002-07-30', num_comments=120)
G.add_node('User4', dob='2003-05-30', num_comments=220)
G.add_node('User5', dob='2003-03-13', num_comments=360)

# Add edges to the graph
G.add_edge('User1', 'User2')
G.add_edge('User2', 'User3')
G.add_edge('User3','User1')
G.add_edge('User2','User4')
G.add_edge('User3','User4')
G.add_edge('User4','User5')

# Find the user with maximum friends
max_friends = max(G.degree, key=lambda x: x[1])[0]
print(f"The user with maximum friends is {max_friends}")

# Find the user with maximum comments
max_comments = max(G.nodes, key=lambda x: G.nodes[x]['num_comments'])
print(f"The user with maximum comments is {max_comments}")

# Find the user with minimum comments
min_comments = min(G.nodes, key=lambda x: G.nodes[x]['num_comments'])
print(f"The user with minimum comments is {min_comments}")

# Find users with birthdays in this month
import datetime
current_month = datetime.datetime.now().month
birthday_users = [
    node for node in G.nodes if int(G.nodes[node]['dob'].split('-')[1]) ==
current_month
    ]
print(f"Users with birthdays in this month are {birthday_users}")
```

Output:

```
The user with maximum friends is User2
The user with maximum comments is User5
The user with minimum comments is User2
Users with birthdays in this month are ['User2', 'User5']
```

Conclusion: Thus, I've successfully completed and performed graph operations on the 'Friends network on Facebook' Model. I've successfully computed a user with max. friends & users with min. & max. comments.