Assignment on :-

Finding Mutual Friends on Social Media

Discrete Mathematics (SC205)

Munjapara Dharmil (202201440)
Avinash Bariaya (202201211)
Dave Preet (202201072)
Kavan Kansodariya (202201223)
Vegad Rutvik (202201143)
Rajan Moradiya (202201063)
Savani Vedant (202201178)

Under the Supervision of

Prof. Name Manish Gupta

Prof. Name Manoj Raut

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

• INTRODUCTION :-

- In this era of internet, out of 8 billion people on earth, near about 60 % population have access to internet. This is a huge number. Nowadays, everybody wants to socialize with each other through different social media application.
- As of recent years, the number of people, active on any social media is about 4.5 billion world-wide.
- Nowadays, users have various options for choosing a social media platform. Many big social media giants, like Instagram, Facebook, Twitter, Whats-app, Snapchat, have active monthly users, on an average, above 2 billion, which plays a significant role for this tech giants.
- So there is a huge competition among these social media companies to survive in the market, as people can easily switch to other social media applications. Thus, social media companies have to keep updating their applications and algorithms, so that people keep using their applications

Figure 1:- Era of Internet

- The research given by TheGuardian shows that the number of global monthly users is predicted to fall by nearly 4 % next year and 5 % in 2024 – more than 32 million in total, after the changes made by new CEO Elon musk, which includes the charges required for verified blue tick, etc.
- This indicates that small updates or changes in any social media application could lead to a sudden spike in active users.

Figure 2:- The Social network

- **MOTIVATION** :-

  - On studying all prime features of all social media applications, one common feature found in all of them was the concept of **Mutual friends.**
  - Companies like Facebook, Twitter, Instagram, etc shows the mutual friends of the one we know. This became a key concept, as it helps the social media in many ways.
  - When social media platforms display mutual friends, it enables users to discover and connect with people they may have common acquaintances with, facilitating new connections and expanding their social network. It also increase the trust and credibility of user on given social media application, as it give the sense of familiarity while using it.
  - Thus, we decided to identify the mutual friends using the knowledge of SC205 (Discrete Mathematics), by using the concept of graph theory.

- **Graph theory** :-

– Graph theory is an key concept for finding mutual friends between two users. Graph theory mainly consists of nodes and edges , which represents the relation between two users.

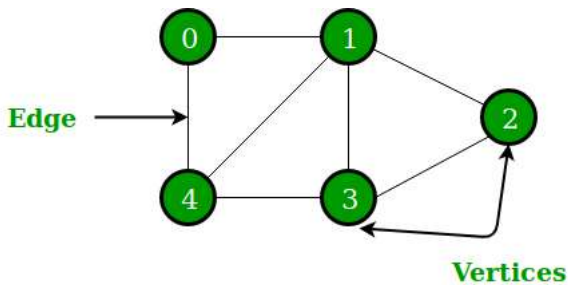


Figure 3:- Basic Graph

– In graph, every node is represented by the active users on a particular social media, and the edge between two users indicates that both users know each other.

– Thus, by indicating all friends by using nodes and edges, it will be easy to determine the mutual friends.

– For exploring all nodes of the user, we have to use one concept of graph theory known as **Breadth First Search**

- **BFS** :-

– **What is BFS:-**

* In BFS(**Binary first search**), we will explore all the nodes from starting node, by using the concept of queue.
* Here is an step wise Algorithm for Breadth first search
  - Choose a starting vertex: Select a vertex from the graph as the starting point for the BFS traversal.
  - Enqueue the starting vertex: Place the starting vertex into a queue data structure. This queue will hold the vertices that are waiting to be processed.
  - Mark the starting vertex as visited: Mark the starting vertex as visited to keep track of which vertices have been explored.
  - Start the BFS traversal :
  - Dequeue a vertex from the queue: Remove a vertex from the front of the queue and consider it for further processing.
  - Process the vertex: Perform any desired operations on the dequeued vertex (e.g., print its value).
  - Enqueue the unvisited neighboring vertices: Look at all the adjacent vertices of the dequeued vertex that have not been visited. Enqueue these unvisited vertices into the queue and mark them as visited.
  - Repeat these steps until the queue becomes empty.
  - Continue the BFS traversal: If there are remaining unvisited vertices in the graph, select a new starting vertex (if desired)

and repeat steps 2 to 4. This ensures that all the vertices in the graph are visited.

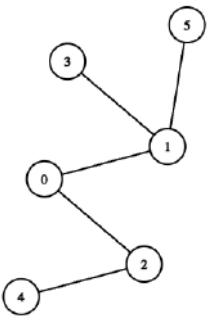


Figure 4:- Graph Example

What is Queue:-

* Queue is one the important linear data structure, in which the addition of data follows certain type of rules.
* Queue follows **FIFO(FIRST IN FIRST OUT)** rule.
* In this, the element which is entered last in the memory will also be popped up last, and the element which is entered first will be popped up first.

Figure 5:- Pictorial example of Queue

Why we use BFS:-

* For identifying the mutual friends from account, we need to explore each and every node linked to head node **(Your account)**
* If the number of nodes linked to your account are in single digit, then it would be easy to explore all other nodes, without any algorithm.
* According to the data, the average number of friends on facebook was reported as 338, going up to million, if you are a celebrity.
* So for dealing large number of nodes**(accounts)**, we need to use an algorithm, to explore all nodes appropriately.
* That's why we use BFS to find the mutual connection on social media.

Alternate to BFS:-

* We can also use other algorithm known as **Depth First Search(DFS)**, which is a kind of similar to BFS.
* In DFS, when we reach any child node of user, then we will not go to new child node of user, until we explore all sub node of first child node.
* After exploring all sub nodes of given child node, then we will move to other child node of parent node.

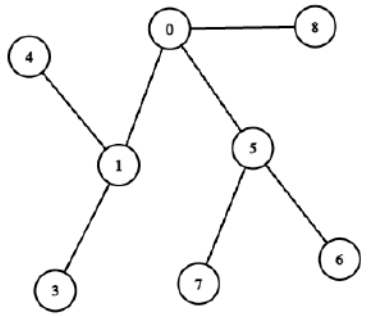


Figure 6:- Graph

∗ For above graph, both the path, BSF and DSF are stated below.

- BFS :- { 0,1,5,8,4,3,7,6 }
- DFS :- { 0,1,4,3,5,7,6,8 }

- **Implementing the mathematical model :-**

– Let's take an example of having 9 friends on particular social media application. Let's represent the node 1 as the main person, and other 9 person as different node, with the linkage between them as shown in below figure.

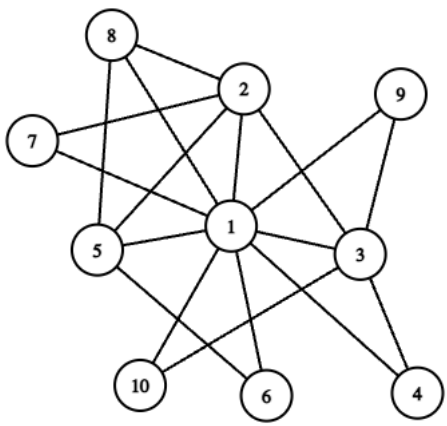


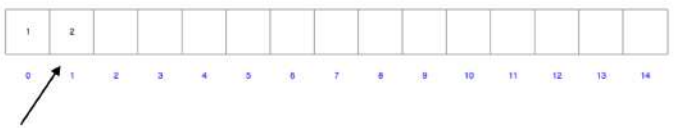
Figure 7:- Example of social media account.

– Starting from the node 1, Add element 1 to queue and start exploring any child node, step wise.

  * Node 2:-

    · Create a graph starting from Node 1 and show a linkage between 1 and 2.

    · Add 1 and 2 in Queue.



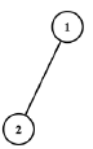
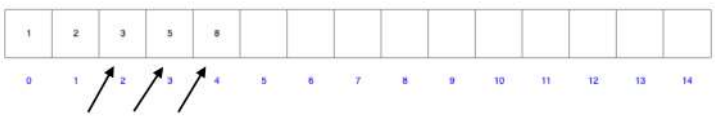


Figure 8:- Initial graph and Queue

    · Now, explore adjacent nodes of nodes 2, and add it to queue and graph

· Adjacent to node (2), there exist, node (3) and node (8). Add (3) and (8) to queue, and add it to graph.



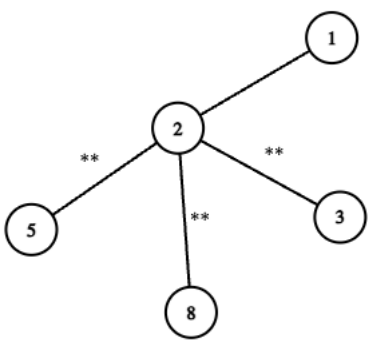


Figure 9:- In second step

Since (8) and (3) are also friends of node (1), we will put a sign

of double star, on the graph.

* By this way, we can track the record of common friends, we wanted to short-list.
* Since, all the nodes connected to (2) have been explored, we will delete (2) and (1) from queue.

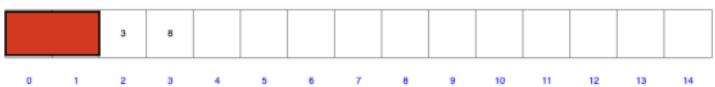


Figure 10:- Queue

* Now starting from node (5), we know that it was already marked, when explored for node (2), so we will directly add (6) to the queue, and to node (5).

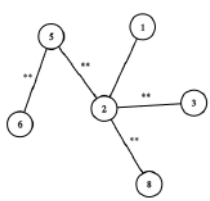


Figure 12:- Graph

* By this way, when we will explore all, by using this algorithm, we will left with an empty queue, and finally having graph similar to given graph.

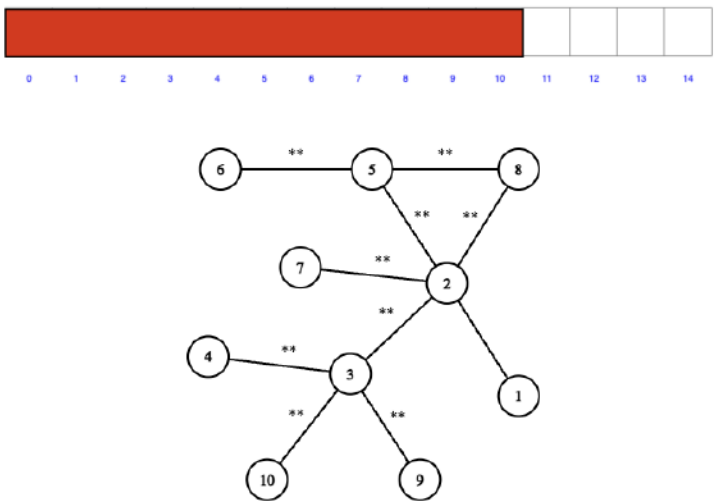


Figure 13:- Final Queue and Graph

- Now, observing the graph constructed from above example, we can differentiate the edges marked with double star sign to conclude the model.
- The set of mutual friend is :-
  { (2,5) , (2,8) , (5,6) , (2,7) , (2,3) , (3,4) , (3,10) , (3,9) }

- **Conclusion :-**

  - From the above algorithm, we can find all mutual friends by using graph theory, and the concept of **Breadth First Search**.
  - If in our graph, there are **(E)** number of edges and **(N)** number of nodes, then the time complexity for exploring all nodes would be equal to :- **O(N + E) {Big O of the value N + E}**