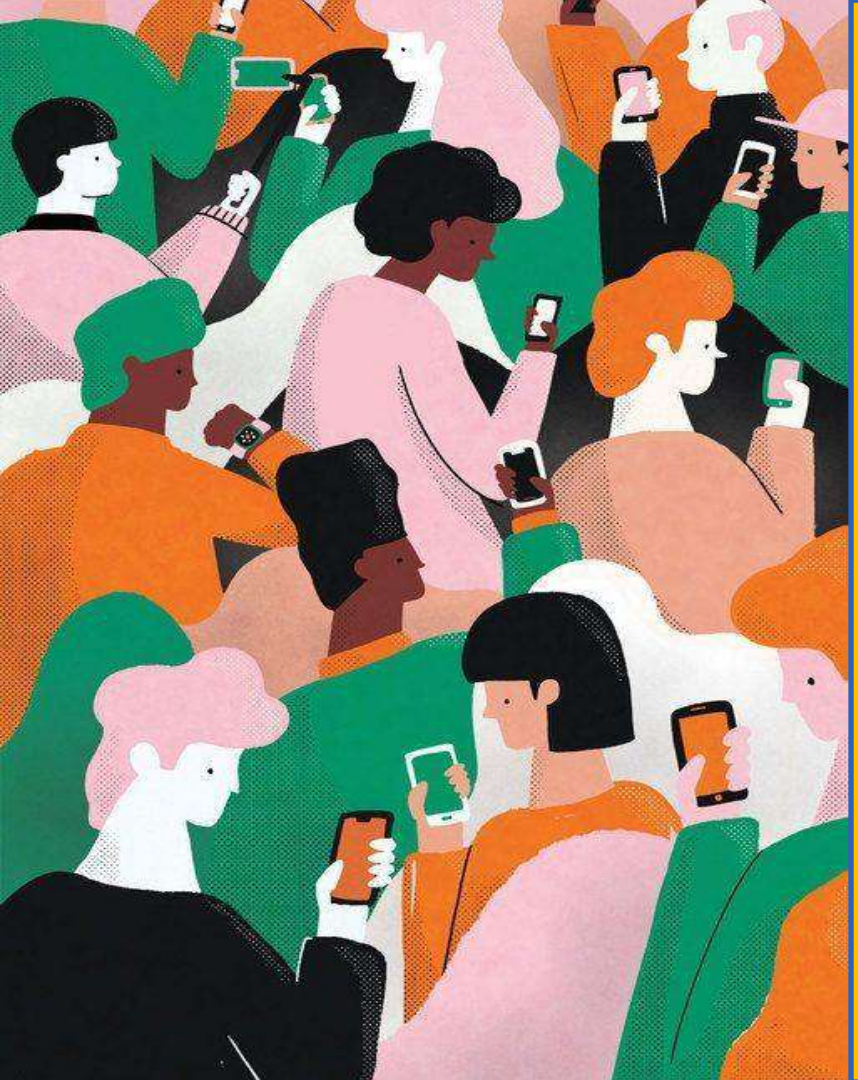


MUTUAL FRIENDS ON SOCIAL MEDIA

Prof. Manish K. Gupta
Prof. Manoj Raut

Group Members:-

Munjapara Dharmil(202201440)

Dave Preet (202201072)

Vegad Rutvik (202201143)

Baraiya Avinash(202201211)

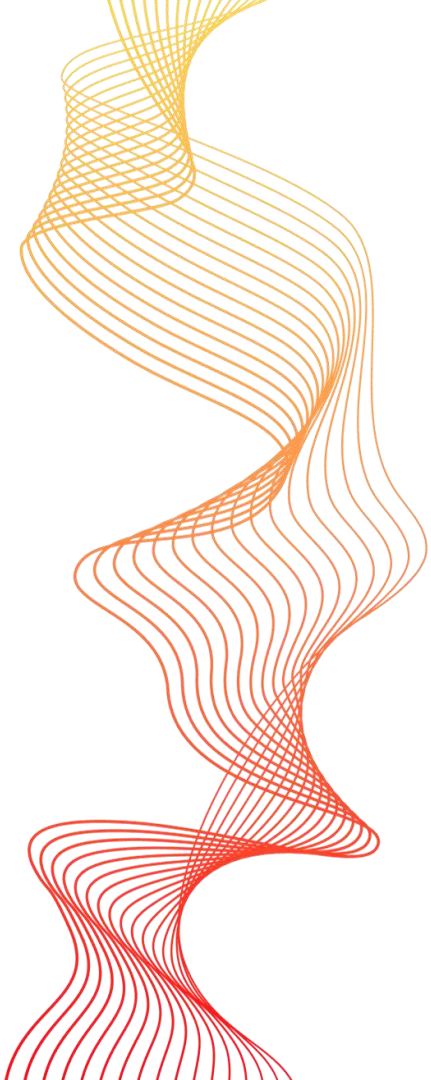
Kansodariya Kavan (202201223)

Savani Vedant (202201178)

Moradiya Rajan(202201063)

Introduction

- Approximately 60% of the world's population has internet access, with about 4.5 billion people being active on social media. Major platforms like Instagram, Facebook, Twitter, WhatsApp, and Snapchat have more than 2 billion active monthly users each.
- The fierce competition among these companies requires them to constantly update their applications and algorithms to retain users. small updates or modifications to social media platforms can have a significant impact on user engagement and overall active user numbers.

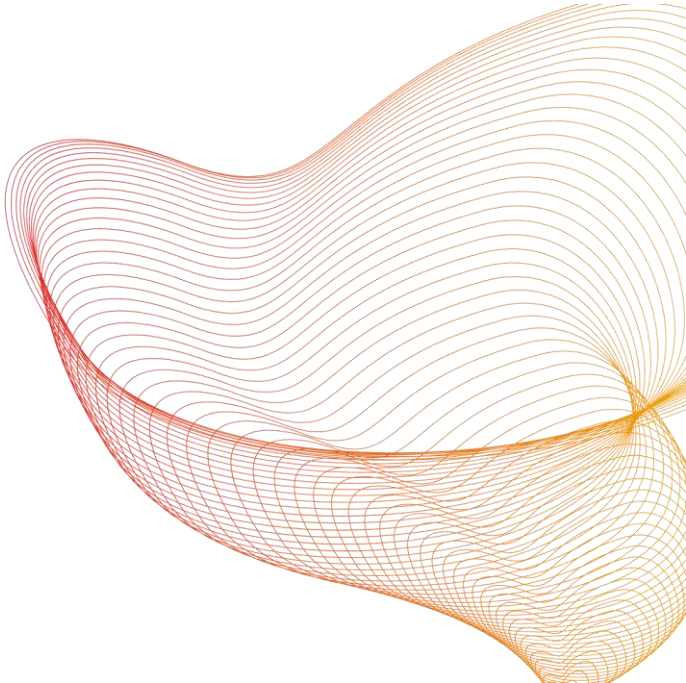


Motivation behind this topic

- 60% of the world's population has access to the internet.
- 4.5 billion people are active on social media worldwide.
- Instagram, Facebook, Twitter, WhatsApp, Snapchat, have active monthly users above 2 billion on average.
- Competition is high in the social media market, requiring constant updates and changes to applications.

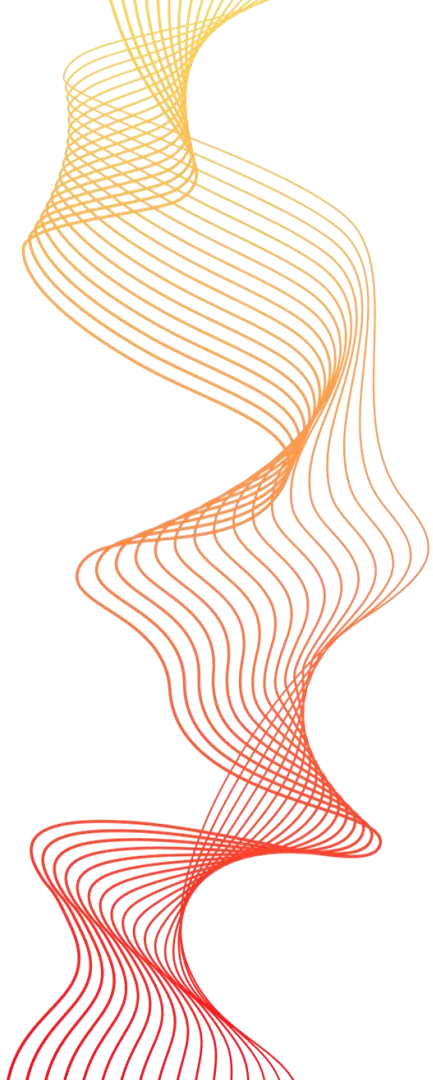
Use of Graph Theory

- With help of graph theory we can implement this topic accuratly.
- Graph theory is a key concept for finding mutual friends between two users on social media.
- Consists of nodes and edges that represent the relationships between users.
- Every node represents an active user, and the edge between two users indicates that they know each other.
- By indicating all friends using nodes and edges, it becomes easy to determine mutual friends.



Concept of Queue

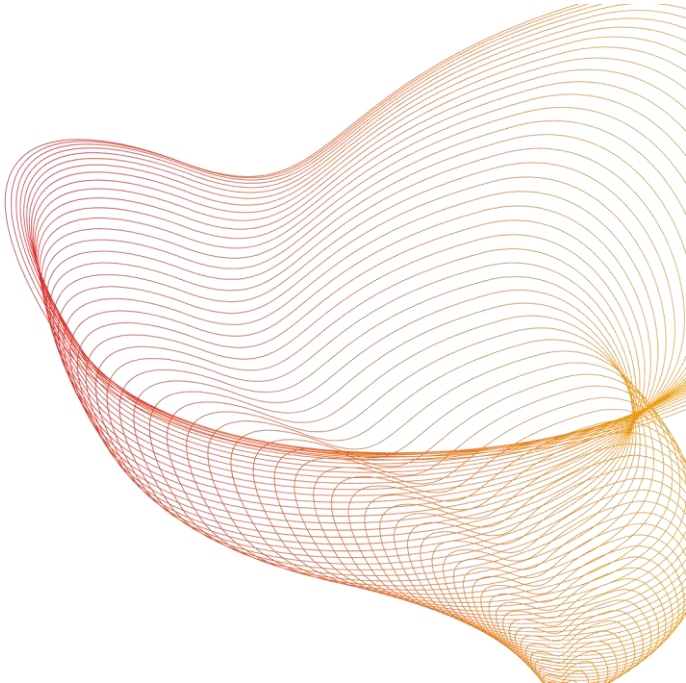
- Queue is one the important linear data structure, in which the addition of data follows certain type of rules.
- Queue follows FIFO(FIRST IN FIRST OUT) rule.
- In this, the element which is entered last in the memory will also be popped up last, and the element which is entered first will be popped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



Concepts of Graphs

- For detecting mutual friends we have major two concepts :

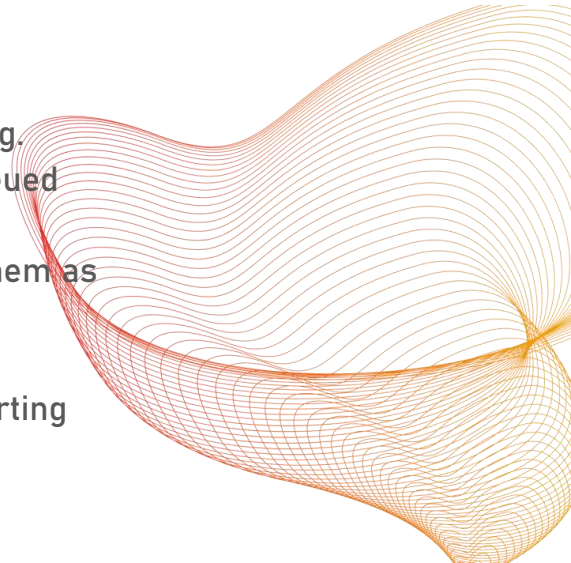
1. BFS(Breadth First Search)
2. DFS(Depth First Search)



Breadth First Search(BFS)

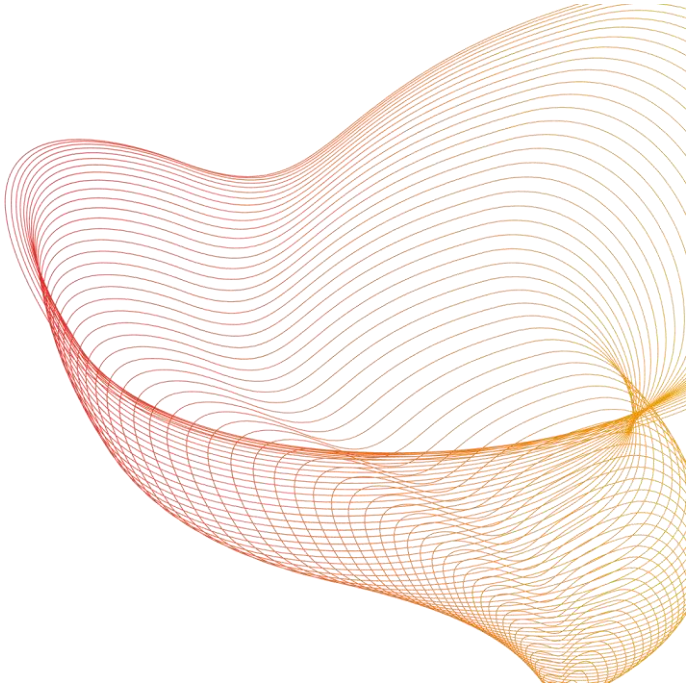
- BFS is a way to explore all nodes from a starting node using the concept of a queue.
- Algorithm for BFS:

1. Choose a starting vertex.
2. Enqueue the starting vertex.
3. Mark the starting vertex as visited.
4. Dequeue a vertex from the queue and consider it for further processing.
5. Process the vertex by performing any desired operations on the dequeued vertex.
6. Enqueue the unvisited neighboring vertices into the queue and mark them as visited.
7. Repeat these steps until the queue becomes empty.
8. If there are remaining unvisited vertices in the graph, select a new starting vertex and repeat steps 2 to 4.



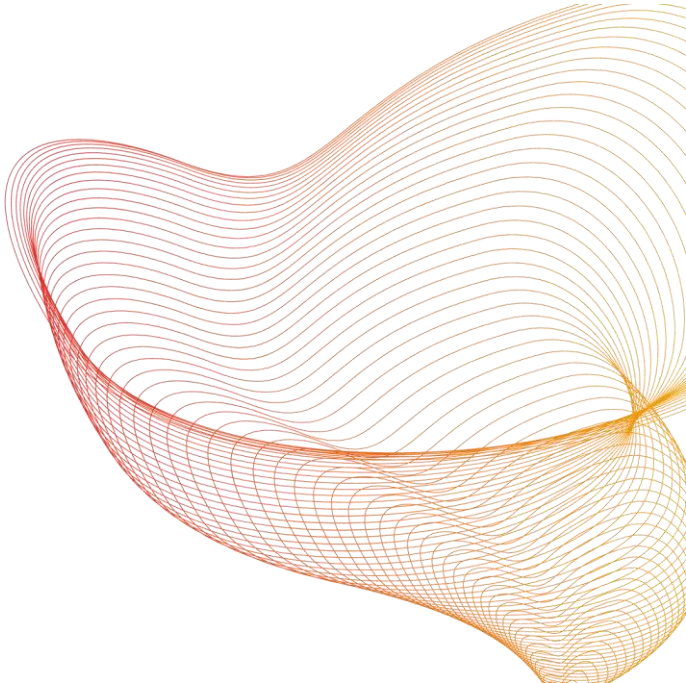
Why use BFS for finding mutual friends on social media?

- We need to explore each and every node linked to the head node (your account) for identifying mutual friends.
- Using BFS, an algorithm can efficiently explore all nodes in a large dataset to find mutual connection.



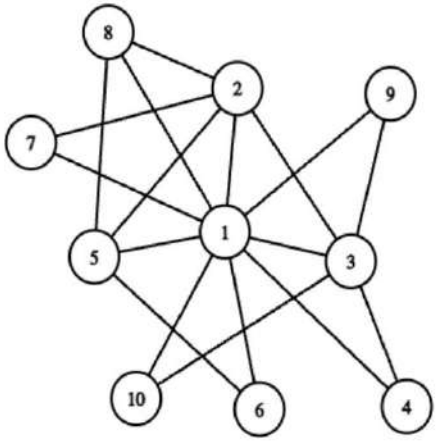
BFS vs DFS

- DFS (depth first search) is another way to traverse a graph, where unvisited nodes are explored as deeply as possible before backtracking.
- BFS is more suitable for finding the shortest path between two nodes.
- DFS is more suitable for applications such as finding a path between two nodes that satisfies certain criteria.



Real life example of BFS

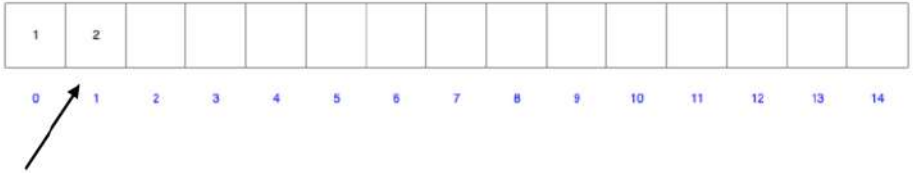
- Let's take an example of having 9 friends on particular social media application.
Let's represent the node 1 as the main person, and other 9 person as different node, with the linkage between them.
- Starting from the node 1, Add element 1 to queue and start exploring any child node, step wise.

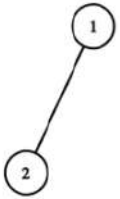


- Node 2:-

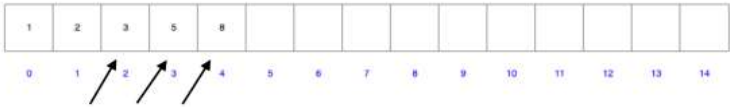
· Create a graph starting from Node 1 and show a linkage between 1 and 2.

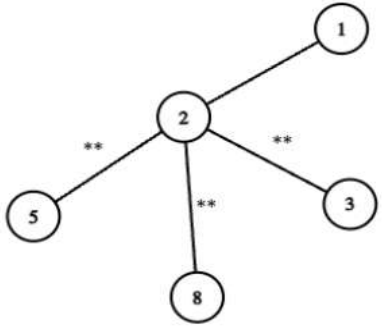
· Add 1 and 2 in Queue



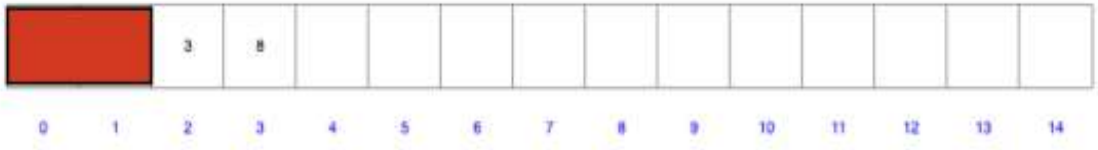


Now, explore adjacent nodes of nodes 2, and add it to queue and graph.
Adjacent to node (2), there exist, node (3) , node (5) and node (8).
Add (3) , (5) and (8) to queue, and add it to graph.

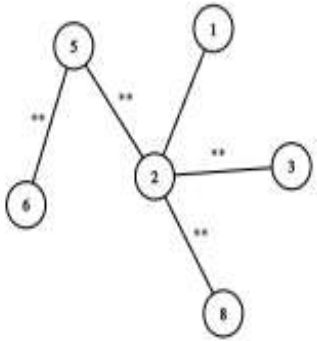




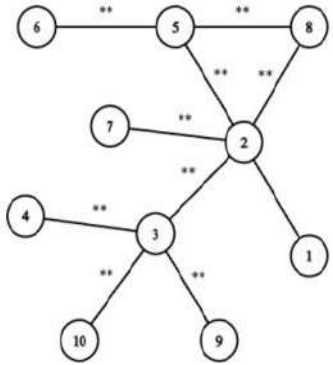
Since (8) and (3) are also friends of node (1), we will put a sign of double star, on the graph.
By this way, we can track the record of common friends, we wanted to short-list.
Since, all the nodes connected to (2) have been explored, we will delete (2) and (1) from queue



- 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).



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.

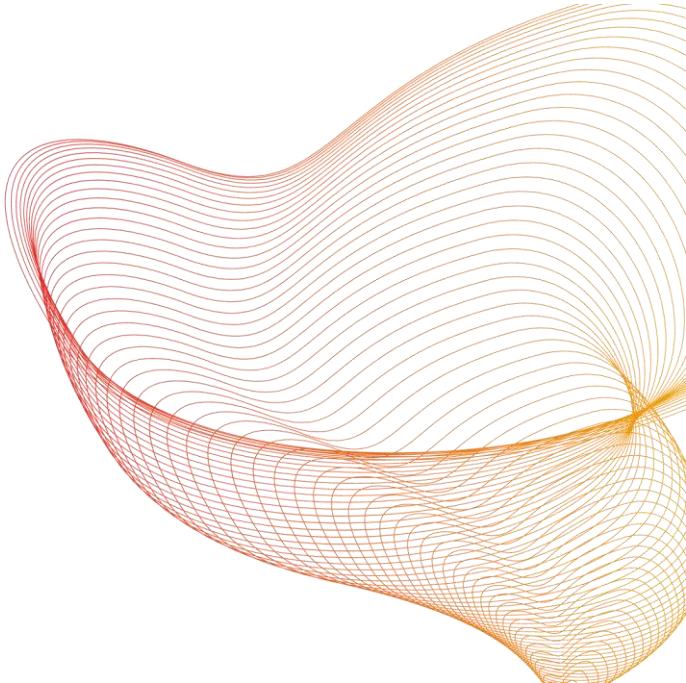


Deriving the Set of Mutual Friends

- Observe the graph constructed from the example, and differentiate edges marked with a double-star sign.

- The set of mutual friends for the example is :-

{ (2,5) , (2,8) , (5,6) , (2,7) , (2,3) , (3,4) , (3,10) , (3,9) }.



Conclusion and Time Complexity

- By using BFS in graph theory, we can efficiently find all mutual friends.
- The time complexity for exploring all nodes in the graph using BFS is O(N + E), where N is the number of nodes and E is the number of edges.