NAT NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

(KARACHI CAMPUS)

FAST School of Computing

**Project Report**

**Data Structure**

**Friend’s Recommendation system**

**Group Members: Section: F**

**Hamza Iqbal** (21k-4513)

**Hussain Malik** (21k-4576)

**Awesh Kumar** (21k-4526)

***Introduction:***

Social media platforms use graph to represent relations between it's users. Our project is inspired from that idea and we have simulated something similar to that.

***Solution Design:***

Just like Facebook: "people you may know", our project recommends friends to a user. There are basically 10 users, and a graph to show connections between them. There are various options to manipulate a user's data(liking posts, sending and accepting friend requests etc) which then affect the graph. The graph basically represents weight of the connection between all the people with each other, and the weight is affected by how many mutual friends two people have, and how many common friends they both have.

Greater the weight closer the friendship level.

Every user will have an option to see his friends in a descending order of weight, closest friend on the top and least close at the bottom.

Finally, the user will get friends recommendations on a criteria: people who are friends of the closest friend of the user, and not friends directly with the user, will get shown in the people you may know.

***Functions And Classes :***

Classes:

Class QueueNode:

Functions:

Constructor:

QueueNode()

QueueNode(int p, int r)

Class QueueP

Functions:

void insert(int p, int r)

int top()

void display()

void dequeue()

Class Vertes:

Functions:

void NodeInfo()

Class Graph:

Functions:

void updateWeight(

void calculateClosest(int n1)

void displayGraph()

int calWeight(int i, int j)

void addFriend(int Person

void acceptRequest(int Person )

void showFriendshipLevel(int person)

void showMutuals(int person)

***Implementation details:***

We have used various data structures to store and manipulate data. There is a vertex class which stores a string variable, two vector objects.

Then there is a gaph of these 10 vertices which shows weight and have 8-9 functions to manipulate that data.

Finally we have a priority queue in which the element with hightest weight goes

above and stay space.

***Conclusion:***

This project taught us how data structures are the backbone of any code. And it gave us insights into how real world problems can be solved using code.