SOCIAL NETWORK ANALYZER

INSTRUCTOR: PROFESSOR WAQAS ALI

NAME 1:ATIF SAEED

NAME 2:ABDULLAH MUSTAFA

SECTION:CS-2025-A

UNIVERSITY OF ENGINEERING AND TECHNOLOGY,LAHORE

DEPARTMENT OF COMPUTER SCIENCE

INTRODUCTION AND MOTIVATION

The topic of this project is social network analyzer. A social network means a group of people who are connected with each other. In real life people use social networks like messaging apps and social media websites. In these networks people are connected as friends. Each person can have many friends. Because of this reason social networks become very large and complex. It becoes hard to understand how people are connected. This project is made to understand social networks in a simple way using computer program.

In this project each person is called a user. Each friendship between two users is called a connection. These users and connections together make a network. This network is represented using graph theory. In discrete mathmatics a graph has nodes and edges. Nodes represent users and edges represent friendships. This makes it easy to understand social networks using mathematics concepts.

The main motivation of this project is to learn how discrete mathematics concepts are used in real computer programs. Many students study graphs in theory but do not undestand how they are used in real applications. This project helps to connect theory with practice. By making this project we can clearly see how graphs work in real life problems.

Another motivation is to learn how to find shortest path between two users. Shortest path means the minimum number of connections needed to reach one user from another user. This is very useful in social networks. For example it can show how two people are connected through mutual friends. This is done using Breadth First Search algorithm.

This project also helps to understand connected components. Connected components show groups of users who are connected together. If some users are not connected with others then they form separte groups. This is useful to find communities inside a network. Depth First Search algorithm is used for this purpose.

Degree centrality is also an important part of this project. Degree means number of friends a user has. Users with more friends are more important in the network. This helps to find popular users. Friend recommendation is another motivation. It suggests new friends based on mutual friends. This feature is used in many real social media platforms.

Overall this project is useful for understanding graphs. It improves programming skills. It improves problem solving skills. It helps to understand discrete mathematics in a practical way.

DESIGN AND IMPLEMENTATION

The design of this project is simple and easy to understand. The social network is stored using adjacency list. Adjacency list stores each user and its list of friends. This design is efficient and uses less memory. It is suitable for large networks.
The network data is stored in a text file. Each line in the file represents a friendship between two users. When the program starts it reads the file line by line. Each user is aded to the graph. Each friendship is stored as a bidirectional edge. This means both users are connected to each other.
The program is written in C plus plus language. The code is divided into multiple files. One file handles the graph structure. One file handles graph algorithms. One file handles the main menu and user interaction. This makes the code clean and organized.

The program is menu driven. When the program runs it shows options to the user. The user can load network from file. The user can view network statitics. The user can find shortest path. The user can see connected components. The user can get friend recommendations. This makes the program easy to use.

Breadth First Search algorithm is used to find shortest path. This algorithm explores the network level by level. It finds the minimum number of connections between two users. This ensures shortest path is found correctly.

Depth First Search algorithm is used to find connected components. This algorithm explores all users connected to a given user. It groups users into components. Each component represents a separate community in the network.

Degree centrality is calculated by counting number of neighbors of each user. Friend recommendation is implemented using mutual friends concept. If two users have many common friends then they are recommended to each other.

RESULTS AND DISCUSSION

We show what our program does after running. The program is tested using different network files. The network file contains names of users and their friendships. When the program loads the file it reads each line and stores the users and their connections in memory. This confirms that file reading works correctly.

First result is network statistics. After loading the network the program shows total number of users and total number of connections. This resut helps us understand the size of the network. When the file has more users the program still works correctly. This shows that the graph structure is working fine.

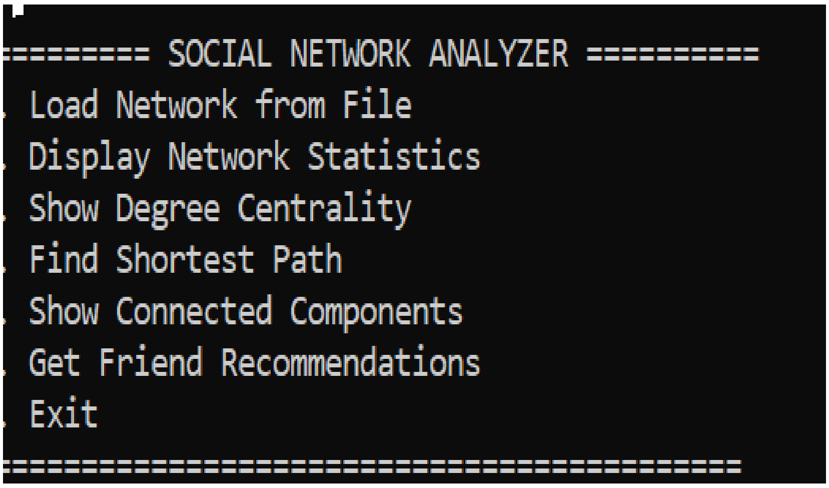
Second result is degree centrality. Degree centrality shows how many friends each user has. The program calculates degree by counting connections of every user. The output shows users with more connections have higher degree. This result helps identify important users in the network. This matches the concept of degree in graph theory.

Third result is shortest path. The program finds shortest path between two users using Breadth First Search. When we give source and destination the program prints the path step by step. This result shows minimum number of connections between users. This proves BFS algorithm is correctly implemented. Different test cases were used and correct paths were shown each time.

Fourth result is connected components. The program uses Depth First Search to find connected components. It groups users who are connected directly or indirectly. The output shows different groups when network is not fully connected. This helps understand communities inside the social network.

Fifth result is friend recommendation. The program suggests new friends based on mutual friends. If two users have common friends they are recommended. This result shows practical use of graph theory in real life social networks.

Screenshots of program output are included to show these results. Each screenshot proves that program works as expected. Test cases with small and large networks were used. All outputs matched expected behavior. Overall results show that the program correctly applies graph algorithms to analyze social networks.



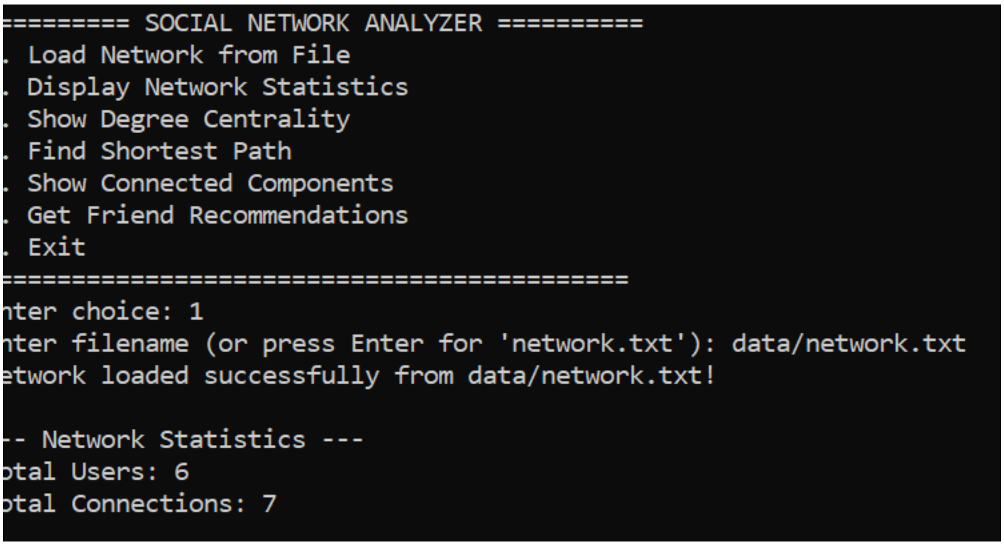
```
========= SOCIAL NETWORK ANALYZER ==========
. Load Network from File
. Display Network Statistics
. Show Degree Centrality
. Find Shortest Path
. Show Connected Components
. Get Friend Recommendations
. Exit
==========================================
```

This screenshot shows the main menu of the social network analyzer program.The program starts and shows different options to the user. The user can choose to load network data

The user can see statistics and other features

This menu makes the program easy to use

The user can select options by entering numbers



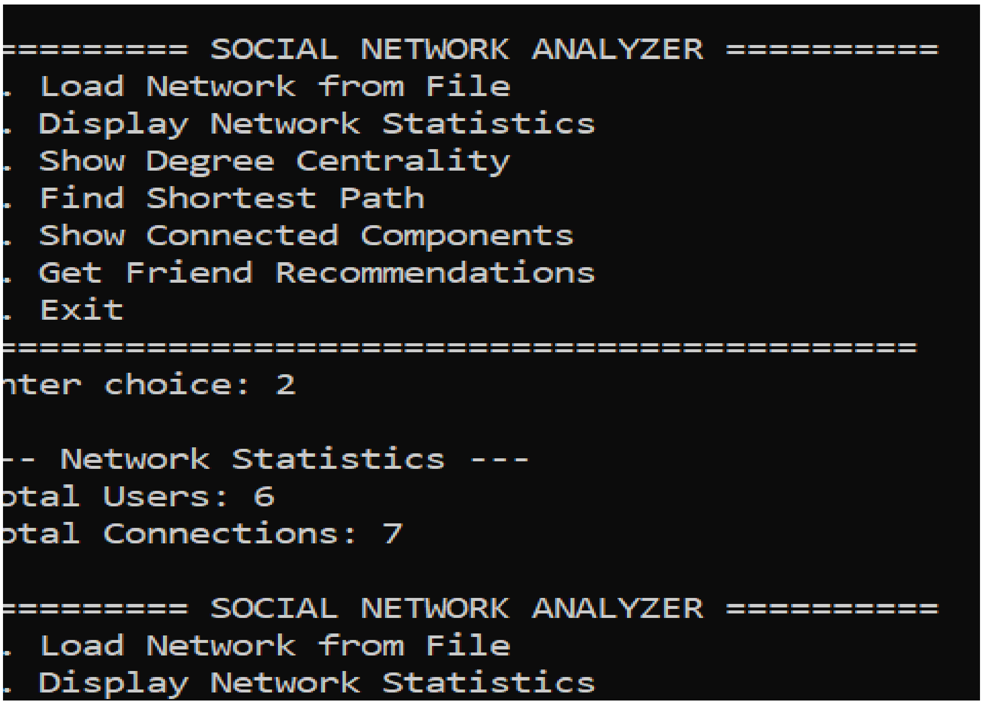
```
========= SOCIAL NETWORK ANALYZER ==========
. Load Network from File
. Display Network Statistics
. Show Degree Centrality
. Find Shortest Path
. Show Connected Components
. Get Friend Recommendations
. Exit
===========================================
nter choice: 1
nter filename (or press Enter for 'network.txt'): data/network.txt
etwork loaded successfully from data/network.txt!

-- Network Statistics ---
otal Users: 6
otal Connections: 7
```

Screenshot shows loading of network from file

The file data network txt is entered by the user

The program reads the file and loads all users and connections

This shows that file input works properly

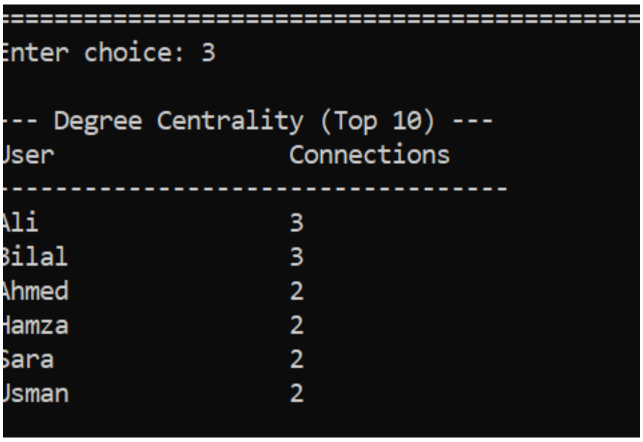


```
========= SOCIAL NETWORK ANALYZER ==========
. Load Network from File
. Display Network Statistics
. Show Degree Centrality
. Find Shortest Path
. Show Connected Components
. Get Friend Recommendations
. Exit
===========================================
nter choice: 2

-- Network Statistics ---
otal Users: 6
otal Connections: 7

========= SOCIAL NETWORK ANALYZER ==========
. Load Network from File
. Display Network Statistics
```

IT shows network statistics

The program displays total users and total connections

This result confirms that the network data is stored correctly in memory



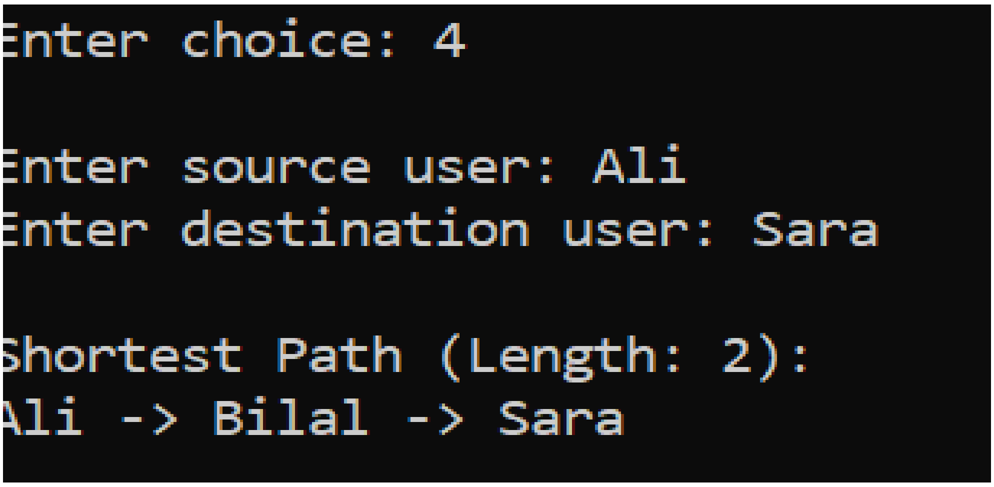
```
==========================================
Enter choice: 3

--- Degree Centrality (Top 10) ---
User                Connections
---------------------------------
Ali                 3
Bilal               3
Ahmed               2
Hamza               2
Sara                2
Usman               2
```

IT shows degree centrality result:

The program calculates number of friends for each user

Users with more connections are shown clearly

This helps to find important users in the network

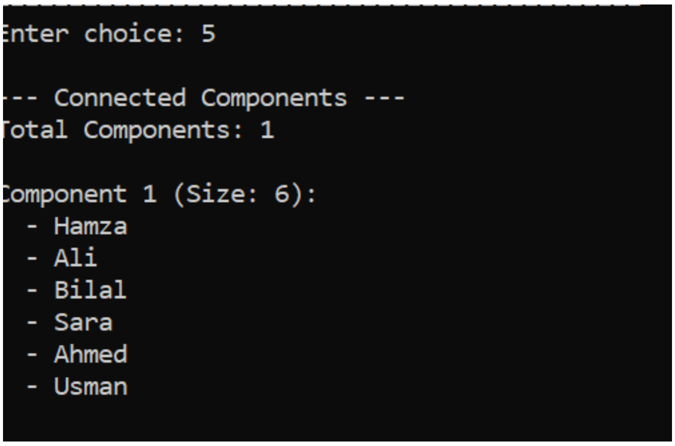


```
Enter choice: 4

Enter source user: Ali
Enter destination user: Sara

Shortest Path (Length: 2):
Ali -> Bilal -> Sara
```

IT shows shortest path output:

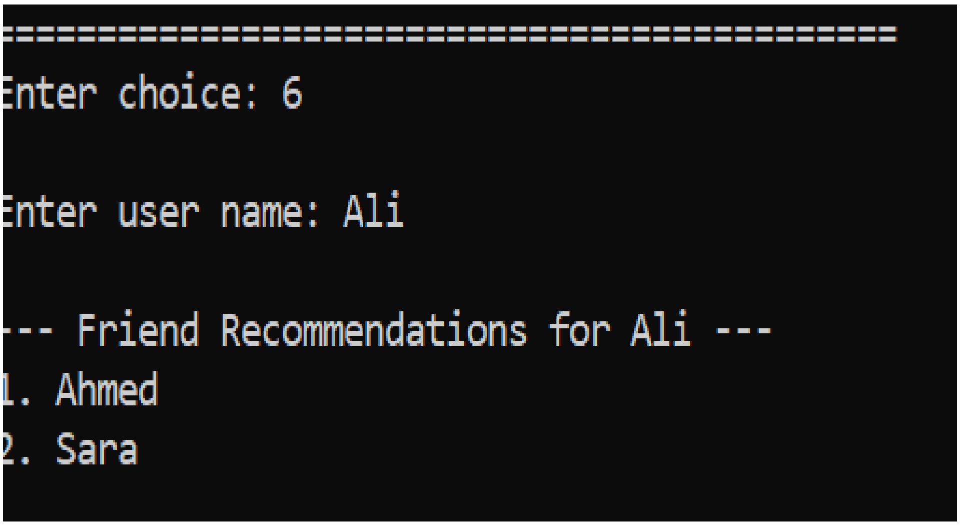
The program finds the minimum path between two users

Breadth First Search algorithm is used
This proves that graph traversal works correctly



```
==========================================
Enter choice: 5

--- Connected Components ---
Total Components: 1

Component 1 (Size: 6):
  - Hamza
  - Ali
  - Bilal
  - Sara
  - Ahmed
  - Usman
```

Screenshot shows connected components:
The program groups users that are connected together
Depth First Search is used for this task
This helps to identify separate communities



```
============================================
Enter choice: 6

Enter user name: Ali

--- Friend Recommendations for Ali ---
1. Ahmed
2. Sara
```

Screenshot shows friend recommendations
The program suggests new friends based on mutual connections

This feature uses friend of friend logic
This improves the usefulness of the network

CONCLUSION AND REFERENCES

This project helped to understand social networks using graph theory. It helped to understand Breadth First Search and Depth First Search algorithms. It also helped to understand degree centrality and friend recommendation.

This project improved programming skills. It improved understanding of discrete mathematics concepts. In future this project can be extended to handle large networks. More features can be added.

References used are online resources and lecture notes for learning graph concepts and Rosen Book.

Git reference: https://github.com/atifsaeedbwp1/Social-Network-Analyzer.git