

Gebze Technical University
Computer Engineering
Department

**CSE 222 – Data Structures And
Algorithms**

Homework-8 Report

Muhammet Akkurt
1901042644

Purpose and Details of the Assignment:

The system is architected into three base classes: Main, which handles user interactions through a console-based interface; Person, which encapsulates the attributes of an individual within the network; and SocialNetworkGraph, which manages the complex relationships between persons using graph-based data structures. These components work together to facilitate the addition and removal of network users, the management of friendships, and the execution of various analytical algorithms that provide insights into the network's structure, such as finding the shortest path between users, suggesting potential friends, and identifying distinct social clusters.

Class Functionality and Explanations:

Main Class : The Main class manages the user interface and interaction with the “SocialNetworkGraph”. The main method of this class makes the system work, initializes the social network with predefined data and performs various operations by processing user input. The “initializeSocialNetwork” method adds predefined contacts and friendships to the network. The “displayMenu” method shows the user the main menu options, while the “getUserOption” method retrieves and validates the user's menu selection. Based on user input, there are several methods that perform operations such as adding and removing contacts, adding and removing friends,

finding the shortest path, suggesting friends and counting clusters.

Person Class: The Person class represents an individual in a social network and stores personal information such as name, age, hobbies and time of joining the network. This class has a constructor method that initializes the person's information, getter and setter methods for each property, a method that formats the timestamp, and a “toString” method that returns a string representing the person.

SocialNetworkGraph: The SocialNetworkGraph class represents a social network graph where the nodes are people and the edges are friendships. This class manages people and their connections and provides methods for various analyses. The “addPerson” method adds a new person with the current timestamp and provides an overloaded version for adding people with a specific timestamp. The “removePerson” method removes a specific person from the network. The “addFriendship” and “removeFriendship” methods add or remove friendships between two people. “findShortestPath” method finds the shortest path between two people using Fuzzy Search (BFS). “suggestFriends” method suggests friends for a person based on mutual friends and common hobbies. “countClusters” method counts and displays the different clusters in the network.

The “findShortestPath” method finds the shortest path between two contacts using BFS. BFS is initialized from the starting contact and the previous contact of each visited contact is followed. When the second contact is reached, the

path is created by following the path from the second contact back to the first contact, resulting in either a path or no path.

The “suggestFriends” method suggests friends for a person based on mutual friends and common hobbies. It iteratively checks all contacts, skips the person and their existing friends, calculates a score based on the number of mutual friends and common hobbies and ranks the suggestions with the highest score.

The “countClusters” method counts and displays the different clusters in the network. All connected contacts are discovered by starting BFS from each unvisited contact and this identifies a new cluster. The person without friends is included in the count. Each BFS scan finds a new cluster and lists the total number of clusters and the members of each cluster.

Test Scenarios:

Start the social network with predefined data.

```
private static void initializeSocialNetwork() {
    try {
        socialNetwork.addPerson(
            name: "Michael Green",
            age: 40,
            Arrays.asList("photography", "traveling", "golf"),
            timestamp: "2024-05-26 10:15:30");
        socialNetwork.addPerson(
            name: "Sarah White",
            age: 22,
            Arrays.asList("gaming", "reading", "music"),
            timestamp: "2024-05-26 11:20:45");
        socialNetwork.addPerson(
            name: "Kevin Black",
            age: 29,
            Arrays.asList("cooking", "traveling", "running"),
            timestamp: "2024-05-26 12:25:50");
        socialNetwork.addPerson(
            name: "Laura Blue",
            age: 35,
            Arrays.asList("painting", "hiking", "swimming"),
            timestamp: "2024-05-26 13:30:00");
        socialNetwork.addPerson(
            name: "Daniel Red",
            age: 28,
            Arrays.asList("running", "music", "photography"),
            timestamp: "2024-05-26 14:35:15");
        socialNetwork.addPerson(
            name: "Olivia Yellow",
            age: 33,
            Arrays.asList("golf", "reading", "yoga"),
            timestamp: "2024-05-26 15:40:20");
        socialNetwork.addPerson(
            name: "James Brown",
            age: 27,
            Arrays.asList("gaming", "music", "cooking"),
            timestamp: "2024-05-26 16:45:25");
        socialNetwork.addPerson(
            name: "Isabella Pink",
            age: 31,
            Arrays.asList("painting", "yoga", "swimming"),
            timestamp: "2024-05-26 17:50:30");
        socialNetwork.addPerson(
            name: "Sophia Green",
            age: 26,
            Arrays.asList("writing", "traveling", "biking"),
            timestamp: "2024-05-26 18:55:35");
        socialNetwork.addPerson(
            name: "Liam Blue",
            age: 32,
            Arrays.asList("photography", "gaming", "cooking"),
            timestamp: "2024-05-26 19:00:40");

        // Adding friendships
        socialNetwork.addFriendship(
            name1: "Michael Green",
            timestamp1: "2024-05-26 10:15:30",
            name2: "Sarah White",
            timestamp2: "2024-05-26 11:20:45");
        socialNetwork.addFriendship(
            name1: "Michael Green",
            timestamp1: "2024-05-26 10:15:30",
            name2: "Kevin Black",
            timestamp2: "2024-05-26 12:25:50");
        socialNetwork.addFriendship(
            name1: "Sarah White",
            timestamp1: "2024-05-26 11:20:45",
            name2: "Laura Blue",
            timestamp2: "2024-05-26 13:30:00");
        socialNetwork.addFriendship(
            name1: "Laura Blue",
            timestamp1: "2024-05-26 13:30:00",
            name2: "Daniel Red",
            timestamp2: "2024-05-26 14:35:15");
        socialNetwork.addFriendship(
            name1: "Daniel Red",
            timestamp1: "2024-05-26 14:35:15",
            name2: "Olivia Yellow",
            timestamp2: "2024-05-26 15:40:20");
        socialNetwork.addFriendship(
            name1: "Olivia Yellow",
            timestamp1: "2024-05-26 15:40:20",
            name2: "James Brown",
            timestamp2: "2024-05-26 16:45:25");
        socialNetwork.addFriendship(
            name1: "James Brown",
            timestamp1: "2024-05-26 16:45:25",
            name2: "Isabella Pink",
            timestamp2: "2024-05-26 17:50:30");
        socialNetwork.addFriendship(
            name1: "Isabella Pink",
            timestamp1: "2024-05-26 17:50:30",
            name2: "Michael Green",
            timestamp2: "2024-05-26 10:15:30");
        socialNetwork.addFriendship(
            name1: "Sophia Green",
            timestamp1: "2024-05-26 18:55:35",
            name2: "Liam Blue",
            timestamp2: "2024-05-26 19:00:40");
        socialNetwork.addFriendship(
            name1: "Sophia Green",
            timestamp1: "2024-05-26 18:55:35",
            name2: "Laura Blue",
            timestamp2: "2024-05-26 13:30:00");
        socialNetwork.addFriendship(
            name1: "Liam Blue",
            timestamp1: "2024-05-26 19:00:40",
            name2: "Kevin Black",
            timestamp2: "2024-05-26 12:25:50");

        // Remove a person
        socialNetwork.removePerson(
            name: "James Brown",
            timestamp: "2024-05-26 16:45:25");
        socialNetwork.removeFriendship(
            name1: "Olivia Yellow",
            timestamp1: "2024-05-26 15:40:20",
            name2: "Daniel Red",
            timestamp2: "2024-05-26 14:35:15");

        // Finding shortest path
        socialNetwork.findShortestPath(
            name1: "Michael Green",
            timestamp1: "2024-05-26 10:15:30",
            name2: "Olivia Yellow",
            timestamp2: "2024-05-26 15:40:20");
        socialNetwork.findShortestPath(
            name1: "Sarah White",
            timestamp1: "2024-05-26 11:20:45",
            name2: "Isabella Pink",
            timestamp2: "2024-05-26 17:50:30");
        socialNetwork.findShortestPath(
            name1: "Sophia Green",
            timestamp1: "2024-05-26 18:55:35",
            name2: "Laura Blue",
            timestamp2: "2024-05-26 13:30:00");

        // Suggesting friends
        socialNetwork.suggestFriends(
            name: "Daniel Red",
            timestamp: "2024-05-26 14:35:15",
            maxSuggestions: 3);
        socialNetwork.suggestFriends(
            name: "Sophia Green",
            timestamp: "2024-05-26 18:55:35",
            maxSuggestions: 2);

        // Counting clusters
        socialNetwork.countClusters();

        // Additional operations to demonstrate more functionality
        socialNetwork.addPerson(
            name: "George Black",
            age: 45,
            Arrays.asList("hiking", "golf", "music"),
            timestamp: "2024-05-26 20:10:50");
        socialNetwork.addFriendship(
            name1: "George Black",
            timestamp1: "2024-05-26 20:10:50",
            name2: "Michael Green",
            timestamp2: "2024-05-26 10:15:30");
        socialNetwork.addFriendship(
            name1: "George Black",
            timestamp1: "2024-05-26 20:10:50",
            name2: "Sophia Green",
            timestamp2: "2024-05-26 18:55:35");

        socialNetwork.findShortestPath(
            name1: "George Black",
            timestamp1: "2024-05-26 20:10:50",
            name2: "Laura Blue",
            timestamp2: "2024-05-26 13:30:00");
        socialNetwork.suggestFriends(
            name: "George Black",
            timestamp: "2024-05-26 20:10:50",
            maxSuggestions: 1);
        socialNetwork.countClusters();
    } catch (Exception e) {
        System.out.println("Error initializing social network: " + e.getMessage());
    }
}
```

Add Person and Friendship:

```
Person added: Michael Green (Timestamp: 2024-05-26 10:15:30)
Person added: Sarah White (Timestamp: 2024-05-26 11:20:45)
Person added: Kevin Black (Timestamp: 2024-05-26 12:25:50)
Person added: Laura Blue (Timestamp: 2024-05-26 13:30:00)
Person added: Daniel Red (Timestamp: 2024-05-26 14:35:15)
Person added: Olivia Yellow (Timestamp: 2024-05-26 15:40:20)
Person added: James Brown (Timestamp: 2024-05-26 16:45:25)
Person added: Isabella Pink (Timestamp: 2024-05-26 17:50:30)
Person added: Sophia Green (Timestamp: 2024-05-26 18:55:35)
Person added: Liam Blue (Timestamp: 2024-05-26 19:00:40)
Friendship added between Michael Green and Sarah White
Friendship added between Michael Green and Kevin Black
Friendship added between Sarah White and Laura Blue
Friendship added between Laura Blue and Daniel Red
Friendship added between Daniel Red and Olivia Yellow
Friendship added between Olivia Yellow and James Brown
Friendship added between James Brown and Isabella Pink
Friendship added between Isabella Pink and Michael Green
Friendship added between Sophia Green and Liam Blue
Friendship added between Sophia Green and Laura Blue
Friendship added between Liam Blue and Kevin Black
```

Remove Person and FriendShip:

```
Person removed: James Brown  
Friendship removed between Olivia Yellow and Daniel Red
```

Find Shortest Path:

```
No path found between Michael Green and Olivia Yellow  
Shortest path: Sarah White -> Michael Green -> Isabella Pink  
Shortest path: Sophia Green -> Laura Blue
```

Suggested friends:

```
Suggested friends for Daniel Red:  
Sarah White (Score: 1.5, 1 mutual friends, 1 common hobbies)  
Sophia Green (Score: 1.0, 1 mutual friends, 0 common hobbies)  
Liam Blue (Score: 0.5, 0 mutual friends, 1 common hobbies)  
Suggested friends for Sophia Green:  
Kevin Black (Score: 1.5, 1 mutual friends, 1 common hobbies)  
Daniel Red (Score: 1.0, 1 mutual friends, 0 common hobbies)
```

Counting Clusters:

```
Counting clusters in the social network...  
Number of clusters found: 2  
Cluster 1:  
Kevin Black  
Michael Green  
Liam Blue  
Sarah White  
Isabella Pink  
Sophia Green  
Laura Blue  
Daniel Red  
Cluster 2:  
Olivia Yellow
```

The rest of the defined data and test case

```
Person added: George Black (Timestamp: 2024-05-26 20:10:50)
Friendship added between George Black and Michael Green
Friendship added between George Black and Sophia Green
Shortest path: George Black -> Sophia Green -> Laura Blue
Suggested friends for George Black:
Laura Blue (Score: 1.5, 1 mutual friends, 1 common hobbies)
Counting clusters in the social network...
Number of clusters found: 2
Cluster 1:
Kevin Black
Michael Green
Liam Blue
Sarah White
Isabella Pink
George Black
Sophia Green
Laura Blue
Daniel Red
Cluster 2:
Olivia Yellow
```

Add Person using the menu (Current system time is assigned as Timestamp):

```
===== Social Network Analysis Menu =====
1. Add person
2. Remove person
3. Add friendship
4. Remove friendship
5. Find shortest path
6. Suggest friends
7. Count clusters
8. Exit
Please select an option: 1
Enter name: John Doe
Enter age: 25
Enter hobbies (comma-separated): hiking,reading,swimming
Person added: John Doe (Timestamp: 2024-05-28 19:38:05)
```