CSE222 Data Structures and Algorithms

Homework 8 Report

In this Project I implemented social network analysis system using graph data structure. Each person in the network is represented as a node, and friendships between them are represented as edges. The project includes various functionalities such as adding and removing people, adding and removing friendships, finding the shortest path between two people, counting clusters of connected people, and suggesting friends based on mutual friends and common hobbies.

Classes

`Person` *Class*

This class represents a single person in the social network system. It contains following variables:

- `name` -> Name of the person
- `age` -> The age of the person
- `hobbies` -> A list of hobbies of the person
- `timestamp` -> The timestamp when the person added to the social network system

It contains following methods:

- `toString` -> Returns a string representation of the person.
- `equals` -> Checks if another object is equal to this person based on name, age, hobbies, and timestamp.
- `getName` -> Returns a name of the person.
- `getAge` -> Returns an age of the person.
- `getHobbies` -> Returns hobbies of that person as a list.
- `getTimestamp` -> Returns a timestamp of that person.

`SocialNetworkGraph` Class

This class represents the social network itself. It contains the following variables:

- `Map<String, Person> people` -> A map that stores people in the network, with a composite key of name and timestamp.

- `Map<Person, List<Person>> friendships` -> A map that stores friendships, where each person is mapped to a list of their friends.

It contains following methods:

- `addPerson` -> Adds a person to the network if they do not already exist.
- `removePerson` -> Removes a person from the network along with their friendships.
- `addFriendship` -> Adds a friendship between two people.
- `removeFriendship` -> Removes a friendship between two people.
- `findShortestPath` -> Finds and prints the shortest path between two people using BFS.
- `suggestFriends` -> Suggests friends for a person based on mutual friends and common hobbies.
    - `calculateFriendshipScore` -> Calculates a "friendship score" for a person based on mutual friends and common hobbies.
    - `countCommonHobbies` -> Counts the number of common hobbies between two people.
    - `countCommmonFriends` -> Counts the number of mutual friends between two people.

- `countClusters` -> Counts and prints the number of clusters in the network using Breadth First Search algorithm.
    - `bfs` -> Finds the shortest path between two people using BFS.

Implementation Details

1. Adding and Removing Person

The `addPerson` method checks for duplicate entries by iterating through the existing people and using the `equals` method of the Person class. If a duplicate is found, it prints a message and returns. Otherwise, it adds the new person to the people map and initializes their friends list in the friendships map.

The `removePerson` method removes the person from the people map and their entry from the friendships map. It also iterates through all friends lists to remove the person from others' friend's lists.

2. Adding and Removing Friendships

The `addFriendship` method checks if both people exist in the network. If they do, it adds each person to the other's friends list in the friendships map. The `removeFriendship` method similarly checks for the existence of both people and removes each person from the other's friends list.

3. Finding Shortest Path

The `findShortestPath` method uses Breadth-First Search (BFS) to find the shortest path between two people. It initializes a queue with the start person and uses a map to keep track of the previous person in the path. If the end person is found, it calls the printPath method to print the path.

4. Counting clusters

The `countClusters` method uses BFS to find all connected components (clusters) in the network. It iterates through all people, performing BFS for each unvisited person to find and count clusters.

- If the person has no friend I count it as single cluster.

5. Suggesting Friends

The `suggestFriends` method suggests friends for a person based on mutual friends and common hobbies. It calculates a friendship score using `calculateFriendshipScore` for each potential friend and sorts them by score. It then prints the top suggestions up to the specified maximum.

Screenshots

```
alperen@DESKTOP-D8VFTCK:/mnt/c/Users/duran/OneDrive/Masaüstü/200104004024_HW_08_ALPEREN_DURAN/src$ make
javac Person.java SocialNetworkGraph.java Main.java
java Main
Person added: John Doe (Age: 25, Hobbies: [reading, hiking, cooking])
Person added: Jane Smith (Age: 22, Hobbies: [swimming, cooking])
Person added: Alice Johnson (Age: 27, Hobbies: [hiking, painting])
Person added: Bob Brown (Age: 30, Hobbies: [reading, swimming])
Person added: Emily Davis (Age: 28, Hobbies: [running, swimming])
Person added: Frank Wilson (Age: 26, Hobbies: [reading, hiking])
Person added: George Martin (Age: 29, Hobbies: [cycling, cooking])
Person added: Hannah White (Age: 24, Hobbies: [reading, running])
Person added: Ian Clark (Age: 31, Hobbies: [swimming, painting])
Person added: Julia Adams (Age: 23, Hobbies: [hiking, running])
Person added: Kyle Baker (Age: 32, Hobbies: [reading, cycling])
Person added: Laura Scott (Age: 26, Hobbies: [swimming, hiking])
Person added: Mike Turner (Age: 28, Hobbies: [running, painting])
Person added: Nina Evans (Age: 30, Hobbies: [cooking, cycling])
Person added: Oscar Young (Age: 27, Hobbies: [reading, swimming])
Person added: Paul Walker (Age: 25, Hobbies: [hiking, running])
Person added: Rachel Hall (Age: 29, Hobbies: [painting, cycling])
Friendship added between John Doe and Jane Smith
Friendship added between John Doe and Alice Johnson
Friendship added between Jane Smith and Bob Brown
Friendship added between Emily Davis and Frank Wilson
Friendship added between George Martin and Hannah White
Friendship added between Ian Clark and Julia Adams
Friendship added between Kyle Baker and Laura Scott
Friendship added between Mike Turner and Nina Evans
Friendship added between Oscar Young and Paul Walker
Friendship added between Rachel Hall and George Martin
Friendship added between Alice Johnson and Hannah White
Friendship added between Frank Wilson and Kyle Baker
Friendship added between Emily Davis and Julia Adams
Friendship added between Mike Turner and Oscar Young
```

```
Shortest path: John Doe -> Jane Smith -> Bob Brown
Shortest path: Emily Davis -> Frank Wilson -> Kyle Baker -> Laura Scott
Shortest path: Mike Turner -> Oscar Young -> Paul Walker
Number of clusters found: 3
Cluster 1:
Emily Davis
Frank Wilson
Julia Adams
Kyle Baker
Ian Clark
Laura Scott
Cluster 2:
John Doe
Jane Smith
Alice Johnson
Bob Brown
Hannah White
George Martin
Rachel Hall
Cluster 3:
Mike Turner
Nina Evans
Oscar Young
Paul Walker
```

Alperen Duran 200104004024