

CSE222 Data Structures and Algorithms Homework 8 Report

(Ahmet Özdemir)

1. Introduction

In this report, I will detail the implementation of the social network analysis system. The project involves analyzing and manipulating a graph data structure to model social networks. I will explain the classes implemented, the overall design approach, and the functionality provided by the system.

2. SocialNetworkGraph.java

The `SocialNetworkGraph` class is the core of the project, representing the social network as a graph where nodes are people and edges represent relationships.

- Node Class: Represents each individual in the network, including their unique ID and connections to other nodes.
- `addPerson(String id)`: Adds a new person to the network with the given ID.
- `removePerson(String id)`: Removes a person and their connections from the network.
- `addConnection(String id1, String id2)`: Establishes a connection between two people in the network.
- `removeConnection(String id1, String id2)`: Removes the connection between two people.
- `getFriends(String id)`: Returns a list of friends for a given person.
- `shortestPath(String id1, String id2)`: Calculates the shortest path between two people using BFS.
- `areConnected(String id1, String id2)`: Checks if two people are connected in the network.

The graph is implemented using an adjacency list for efficient space utilization and quick access to neighbors.

3. Person.java

The `Person` class models an individual in the social network, holding their unique ID and a list of their connections.

- `getID()`: Returns the unique ID of the person.
- `addFriend(Person friend)`: Adds a friend to the person's list of friends.
- `removeFriend(Person friend)`: Removes a friend from the person's list of friends.
- `getFriends()`: Returns the list of the person's friends.

The `Person` class is used within the `SocialNetworkGraph` to represent each node in the graph.

4. Main.java

The `Main` class serves as the entry point of the application, handling user input and executing the corresponding graph operations.

- `main(String[] args)`: Reads commands from an input file and executes them.
- `processCommands(List<String> commands)`: Processes each command to add/remove people, add/remove connections, and query the network.

Commands are read from an input file and processed to perform the respective graph operations. The `Main` class also handles exceptions and ensures the integrity of the graph operations.

5. Generating Random Input

A utility class generates random input commands for testing the social network analysis system. The commands include adding and removing people, and adding and

removing connections between people. This class ensures that the generated commands are valid and can be used to test the system's performance and correctness.

- generateRandomInput(int numCommands): Generates a specified number of random commands.
- generatePersonID(): Creates a unique person ID.
- generateAddPersonCommand(): Generates an 'add person' command.
- generateRemovePersonCommand(): Generates a 'remove person' command.
- generateAddConnectionCommand(): Generates an 'add connection' command.
- generateRemoveConnectionCommand(): Generates a 'remove connection' command.

6. Sample Outputs

Sample outputs will be provided to illustrate the results of various operations on the social network graph. These outputs include the state of the network after each operation, the result of queries, and the paths found between individuals.

7. Usage With Makefile

The `Makefile` is designed to automate the compilation and execution of the project.

- all: Compiles all Java source files.
- clean: Cleans the output directory by removing compiled files.
- run: Runs the main class with the input file.
- build_and_run: Cleans, compiles, and runs the project.
- javadoc: Generates Javadoc documentation.
- generate_input: Generates a random input file.
- full_build_and_run: Performs a full build, documentation generation, and runs the main class.

OUTPUTS:

```
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/2_SINIF/spring/Data-Structures/homework/hw8/HW8_Files/HW8_Demo/src$ make run
===== 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: ahmet
Enter age: 23
Enter hobbies (comma-separated): football,coding,gym,game
Person added: ahmet (Timestamp: 2024-05-29 17:43:39)
===== 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: esma
Enter age: 21
Enter hobbies (comma-separated): volleybol,coding,gym,game,painting
Person added: esma (Timestamp: 2024-05-29 17:44:04)
===== 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: 3
Enter first person's name: ahmet
Enter first person's timestamp (yyyy-MM-dd HH:mm:ss): 2024-05-29 17:43:39
Enter second person's name: esma
Enter second person's timestamp (yyyy-MM-dd HH:mm:ss): 2024-05-29 17:44:04
```

```
6. Suggest friends
7. Count clusters
8. Exit
Please select an option: 3
Enter first person's name: ahmet
Enter first person's timestamp (yyyy-MM-dd HH:mm:ss): 2024-05-29 17:43:39
Enter second person's name: esma
Enter second person's timestamp (yyyy-MM-dd HH:mm:ss): 2024-05-29 17:44:04
Friendship added between ahmet and esma
===== 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: baris
Enter age: 28
Enter hobbies (comma-separated): coding,painting,football
Person added: baris (Timestamp: 2024-05-29 17:45:21)
===== 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: aykut
Enter age: 22
Enter hobbies (comma-separated): football,volleybol,basketball,coding
Person added: aykut (Timestamp: 2024-05-29 17:46:20)
===== 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
```

```
Enter hobbies (comma-separated): football,volleybol,basketball,coding
Person added: aykut (Timestamp: 2024-05-29 17:46:20)
===== 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: 6
Enter person's name: ahmet
Enter person's timestamp (yyyy-MM-dd HH:mm:ss): 2024-05-29 17:43:39
Enter maximum number of friends to suggest: 3
Suggested friends for ahmet:
aykut (Score: 1.0, 0 mutual friends, 2 common hobbies)
baris (Score: 1.0, 0 mutual friends, 2 common hobbies)
===== 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: 5
Enter first person's name: ahmet
Enter second person's name: aykut
No path found between ahmet and aykut
===== 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: 5
Enter first person's name: ahmet
Enter second person's name: esma
```

```
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: 5
Enter first person's name: ahmet
Enter second person's name: esma
Shortest path: ahmet -> esma
===== 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: 7
Counting clusters in the social network...
Cluster 1:
aykut
Cluster 2:
ahmet
esma
Cluster 3:
baris
Number of clusters found: 3
===== 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: 8
Exiting...
ahmete@ahmete-Inspiron-14-5401:~/DERSLER/2_SINIF/spring/Data-Structures/homework/hw8/HW8_Files/HW8_Demo/src$
```