Gebze Technical University

Department of Computer Engineering

CSE222 Spring 2024

Data Structures and Algorithms

Homework #08

Mehmet Mahir Kayadelen, Student no. 210104004252

Explanation of Implemented Functions

1)findShortestPath

Initialization:

-Retrieve the start and end person objects from the people map.

-If either person is not found, print an error message and return.

-Initialize a prev map to store the previous node in the path, a queue for BFS, and a visited set to track visited nodes.

BFS Algorithm:

-Add the start person to the queue and mark as visited.

-While the queue is not empty, perform the following:

-Dequeue the current person.

-If the current person is the end person, call printPath to display the path and return.

-For each neighbor of the current person, if not visited, add to queue, mark as visited, and update the prev map.

-If no path is found after the BFS, print a message indicating no connection between the two persons.

Purpose: The function finds and prints the shortest path between two people in the social network using the BFS algorithm, which is suitable for unweighted graphs like social networks.

2) countClusters

Initialization:

-Initialize a set to track visited nodes and a counter for clusters.

BFS for Clusters:

-Iterate through all people in the network.

-For each unvisited person, increment the cluster counter and perform BFS to explore all connected -people, grouping them into clusters.

-Print the names of all people in each cluster.

Output: After all clusters are identified, print the total number of clusters found.

Purpose: This function identifies and counts the number of clusters (connected components) in the social network, grouping people who are directly or indirectly connected.

3) removeFriendship

Retrieve Person Objects:

Retrieve the person objects for both names from the people map.

Remove Friendship:

-If both persons exist, remove each person from the other's friend list in the friendships map.

-Print a confirmation message if the friendship is removed, otherwise print an error message.

Purpose: This function removes the friendship between two specified people in the social network, ensuring that the relationship is bidirectionally removed.

4) removePerson

Retrieve Person Object:

Retrieve the person object from the people map.

Remove Person:

-If the person exists, remove the person from the friendships map.

-Iterate through all other people's friend lists and remove this person.

-Print a confirmation message if the person is removed, otherwise print an error message.

Purpose: This function removes a person from the social network, including all their friendships, effectively isolating them from the network.

5) suggestFriends

Initialization:

-Retrieve the person object from the people map.

-If the person is not found, print an error message and return.

-Initialize maps to store scores, mutual friends, and common hobbies for each potential friend.

Scoring Potential Friends:

-Iterate through all people in the network, excluding the person and their current friends.

-For each potential friend, count mutual friends and common hobbies.

-Calculate a score based on mutual friends and common hobbies and store the scores in the maps.

Sorting and Suggesting:

-Sort the potential friends by score in descending order.

-Print the top N friend suggestions, including the score, number of mutual friends, and number of common hobbies for each suggested friend.

Purpose: This function suggests potential friends for a person based on the number of mutual friends and common hobbies, aiming to enhance the social experience by recommending likely compatible connections.

Tester.java

Running Screenshots:

```
Person added: Ali Kaya (Age: 25, Hobbies: [kitap okumak, yürüyüş yapmak, yemek pişirmek])
Person added: Ayşe Yılmaz (Age: 22, Hobbies: [yüzmek, yemek pişirmek])
Person added: Mehmet Demir (Age: 27, Hobbies: [yürüyüş yapmak, resim yapmak])
Person added: Fatma Çelik (Age: 30, Hobbies: [kitap okumak, yüzmek])
Person added: Emre Aydın (Age: 28, Hobbies: [koşmak, yüzmek])
Person added: Zeynep Kara (Age: 26, Hobbies: [kitap okumak, yürüyüş yapmak])
Person added: Murat Öz (Age: 29, Hobbies: [yüzmek, yemek pişirmek])
Person added: Elif Doğan (Age: 24, Hobbies: [resim yapmak, koşmak])
Person added: Ahmet Ak (Age: 31, Hobbies: [yürüyüş yapmak, yüzmek])
Person added: Cemre Kır (Age: 23, Hobbies: [yemek pişirmek, kitap okumak])

Friendship added between Ali Kaya and Ayşe Yılmaz
Friendship added between Ali Kaya and Mehmet Demir
Friendship added between Ayşe Yılmaz and Fatma Çelik
Friendship added between Emre Aydın and Zeynep Kara
Friendship added between Murat Öz and Elif Doğan
Friendship added between Ahmet Ak and Cemre Kır
Friendship added between Mehmet Demir and Murat Öz
Friendship added between Ali Kaya and Cemre Kır
Friendship added between Fatma Çelik and Ahmet Ak

Shortest path: [Ali Kaya (Age: 25, Hobbies: [kitap okumak, yürüyüş yapmak, yemek pişirmek]), Ayşe Yılmaz (Age: 22, Hobbies: [yüzmek, yemek pişirmek]), Fatma Çelik (Age: 30, Hobb
ies: [kitap okumak, yüzmek])]

No connection between Emre Aydın and Murat Öz

Cluster 1:
Ali Kaya
Ayşe Yılmaz
Mehmet Demir
Cemre Kır
Fatma Çelik
Murat Öz
Ahmet Ak
Elif Doğan
Cluster 2:
```

```
Emre Aydın
Zeynep Kara
Number of clusters found: 2

Suggested friends for Ali Kaya:
Murat Öz (Score: 1.5, 1 mutual friends, 1 common hobbies)
Fatma Çelik (Score: 1.5, 1 mutual friends, 1 common hobbies)
Ahmet Ak (Score: 1.5, 1 mutual friends, 1 common hobbies)

Suggested friends for Murat Öz:
Ali Kaya (Score: 1.5, 1 mutual friends, 1 common hobbies)
Ayşe Yılmaz (Score: 1.0, 0 mutual friends, 2 common hobbies)
Cemre Kır (Score: 0.5, 0 mutual friends, 1 common hobbies)

Person removed: Zeynep Kara (Age: 26, Hobbies: [kitap okumak, yürüyüş yapmak])

Friendship removed between Ali Kaya and Ayşe Yılmaz

Cluster 1:
Ali Kaya
Mehmet Demir
Cemre Kır
Murat Öz
Ahmet Ak
Elif Doğan
Fatma Çelik
Ayşe Yılmaz
Cluster 2:
Emre Aydın
Number of clusters found: 2

Friendship added between Emre Aydın and Ayşe Yılmaz
Friendship added between Murat Öz and Fatma Çelik
Friendship added between Ahmet Ak and Elif Doğan

Cluster 1:
```

```
Ali Kaya
Mehmet Demir
Cemre Kır
Murat Öz
Ahmet Ak
Elif Doğan
Fatma Çelik
Ayşe Yılmaz
Emre Aydın
Number of clusters found: 1

Suggested friends for Ali Kaya:
Murat Öz (Score: 1.5, 1 mutual friends, 1 common hobbies)
Ahmet Ak (Score: 1.5, 1 mutual friends, 1 common hobbies)
Ayşe Yılmaz (Score: 0.5, 0 mutual friends, 1 common hobbies)

Suggested friends for Emre Aydın:
Fatma Çelik (Score: 1.5, 1 mutual friends, 1 common hobbies)
Murat Öz (Score: 0.5, 0 mutual friends, 1 common hobbies)
Elif Doğan (Score: 0.5, 0 mutual friends, 1 common hobbies)
```

Explanation:

A new instance of SocialNetworkGraph is created.

This object will be used to manage the social network.

Adding People:

The addPerson method is called multiple times to add ten people to the network.

Each person has a name, age, and list of hobbies.

Adding Friendships:

The addFriendship method is called to establish friendships between various pairs of people.

This step demonstrates the creation of connections (edges) between nodes in the graph.

Finding Shortest Path:

The findShortestPath method is called twice to find and print the shortest path between two pairs of people.

This step illustrates the BFS algorithm to find the shortest path in an unweighted graph.

Counting Clusters:

The countClusters method is called to identify and count the number of clusters (connected components) in the network.

This step shows the identification of distinct groups of connected people.

Suggesting Friends:

The suggestFriends method is called twice to suggest friends for two people based on mutual friends and common hobbies.

This step demonstrates the recommendation system.

Removing a Person:

The removePerson method is called to remove a person from the network.

This step tests the removal of a node and its associated edges.

Removing a Friendship:

The removeFriendship method is called to remove a friendship between two people.

This step tests the removal of an edge between two nodes.

Re-Counting Clusters:

The countClusters method is called again to recount clusters after the removal operations.

This step checks how the removal of nodes and edges affects the structure of the network.

Adding More Friendships:

Additional friendships are established by calling the addFriendship method.

This step demonstrates the dynamic nature of the network, where connections can be added at any time.

Final Cluster Count:

The countClusters method is called once more to perform a final count of clusters after all changes.

This step ensures that the network's current state is correctly analyzed.

Suggesting Friends After Changes:

The suggestFriends method is called again to suggest friends for two people based on the updated network.

This step verifies the recommendation system's responsiveness to changes in the network.