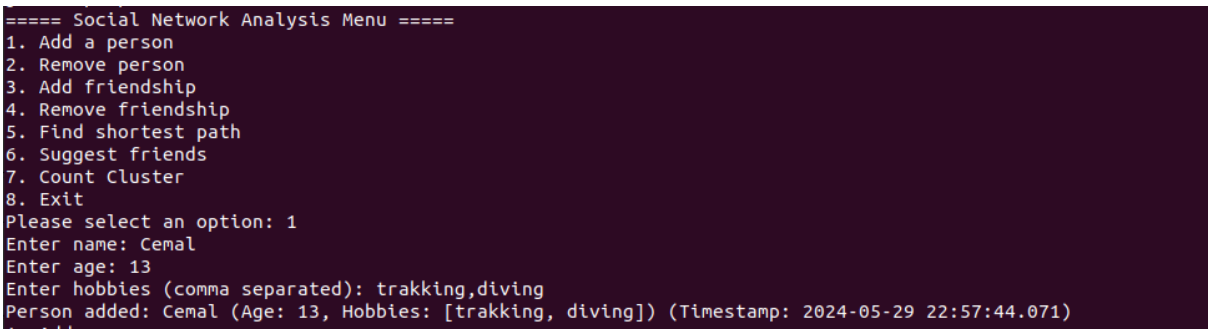
CSE222 / BİL505
Data Structures and Algorithms
Homework #8 – Report
ABDULLAH TÜRKMEN

NOTE:
I initialized these users for test inputs. You can turn off debug mode if you want. Just take it under comment line 18-41 in main.

1. ADD PERSON

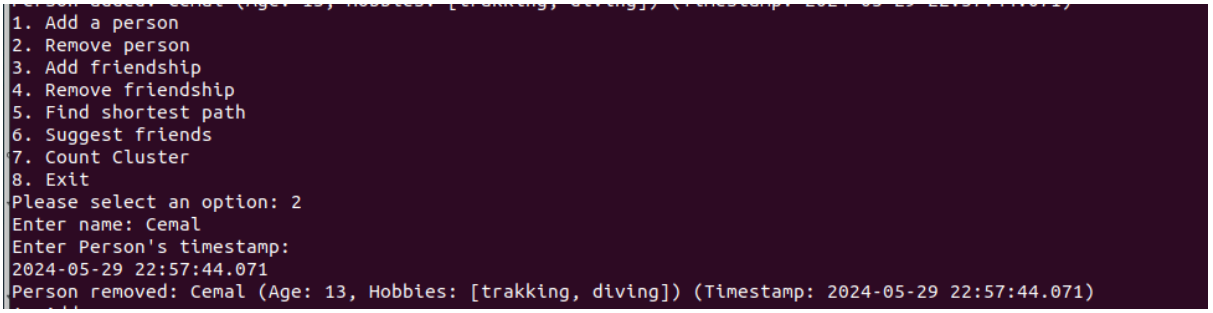
The `addPerson` method adds a new person to the social network. It takes three parameters: `name`, the name of the person to be added, `age`, the age of the person, and `hobbies`, a list of hobbies of the person. The method checks if a person with the same name already exists in the network. If the person already exists, the method prints an error message and returns. If the person does not exist, the method creates a new `Person` object, adds it to the network, and initializes an empty list of friendships for the person. Finally, the method prints a success message to the console.



2. REMOVE PERSON

The `removePerson` method removes a person from the social network. It takes two parameters: `name`, the name of the person to be removed, and `date`, a timestamp to validate the removal. The method retrieves the person from the network and checks if the timestamp is valid. If the timestamp is valid, the method removes the person from the network, removes their

friendships, and updates the friendships of their friends. If the person is found, the method prints a success message to the console and returns the removed person. If the person is not found or the timestamp is invalid, the method prints an error message and returns null.



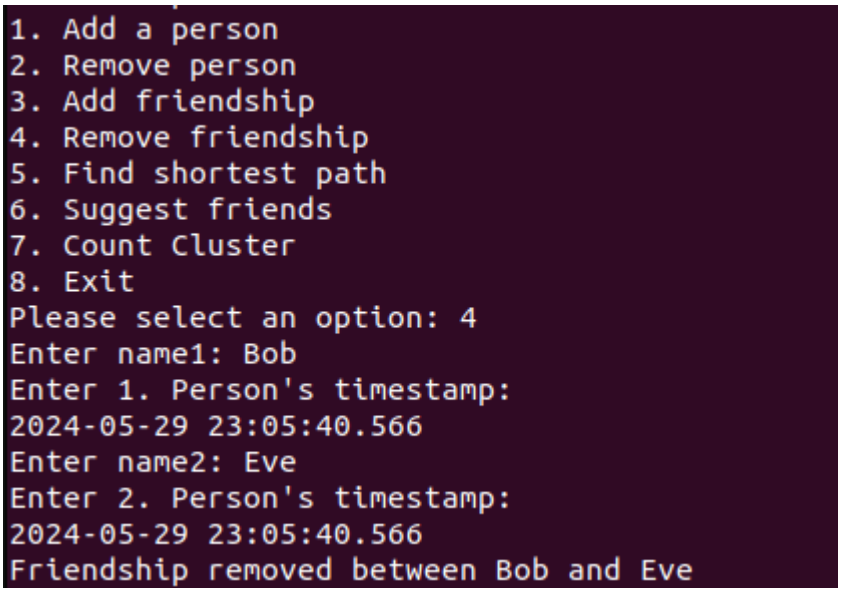
3. ADD FRIENDSHIP

The `addFriendship` method adds a friendship between two people in the social network. It takes four parameters: `name1` and `d1`, the name and timestamp of the first person, and `name2` and `d2`, the name and timestamp of the second person. The method retrieves the two people from the network and adds each other to their respective friend lists. If both people are found, the method prints a success message to the console.

4. REMOVE FRIENDSHIP

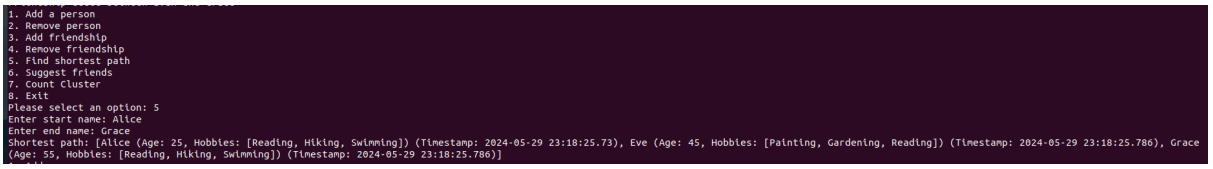
The `removeFriendship` method removes a friendship between two people in the social network. It takes two parameters: `name1` and `name2`, the names of the two people whose friendship is to be removed. The method retrieves the two people from the network and removes each other from their respective friend lists. If both people

are found, the method prints a success message to the console.



5. FIND SHORTEST PATH

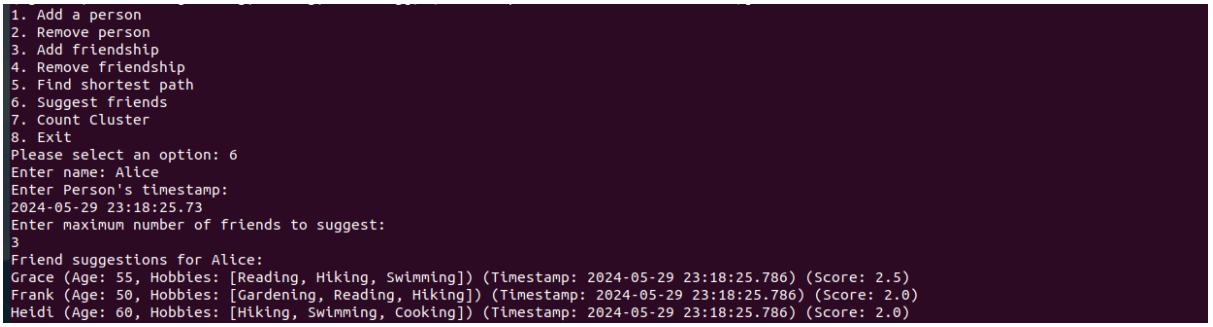
The `findShortestPath` method finds the shortest path between two people in the social network. It takes two parameters, `startName` and `endName`, which are the names of the starting and ending people, respectively. The method uses a breadth-first search (BFS) algorithm to traverse the network and find the shortest path between the two people. If a path is found, it is printed to the console using the `printPath` method. If no path is found, a message is printed indicating that no path exists between the two people.



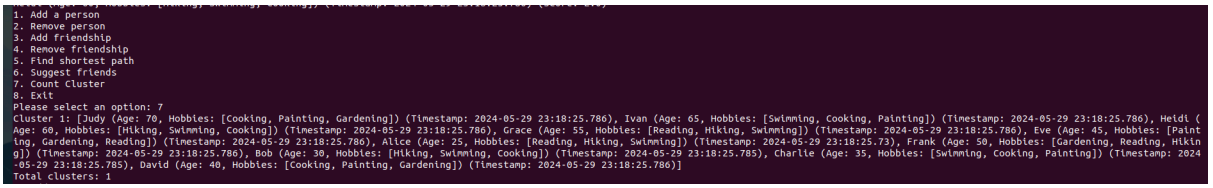
6. SUGGEST FRIEND

The `suggestFriends` method suggests friends for a person based on their mutual friends and hobbies. It takes three parameters: `name`, the name of the person to suggest friends for; `date`, a timestamp to validate the suggestion; and `k`, the number of friend suggestions to return. The method calculates a score for each person in the network based on the number of mutual friends and shared hobbies,

and then returns the top `k` suggestions with the highest scores.



7. COUNT CLUSTER
   The `countClusters` method counts the number of clusters in the social network. A cluster is a group of people who are connected to each other through friendships. The method uses a breadth-first search (BFS) algorithm to traverse the network and identify clusters. It starts by iterating through all people in the network, and for each unvisited person, it performs a BFS to find all connected people and adds them to a cluster. The method then increments a cluster count and prints the cluster to the console. Finally, it prints the total number of clusters found.



==========END OF REPORT==========