

Project: Social Network with Friend Recommendation System

Project Overview:

The Social Network project aims to build a simplified social media system that allows users to connect, explore friendships, and receive friend recommendations based on different graph traversal and searching algorithms. The program will model the social network as an undirected graph, where:

- Users are nodes (vertices), maximum number of users is 100.
- Friendships are edges (connections between users).
- Edges have weights, representing the friendship strength (interaction frequency) in the range [1, 10].

Requirements:

- Read a file containing all users, and their friendships relations. Input example:

6 # number of users (indices from 0 to 5)

Jean # user 0

Alice # user 1

David # user 2

Gregory # user 3

Eric # user 4

Sandra # user 5

4 # number of relations

0 1 4 # Jean is a friend of Alice with strength 4

0 2 6 # Jean is a friend of David with strength 6

2 4 5 # David is a friend of Eric with strength 5

3 5 3 # Gregory is a friend of Sandra with strength 3

Requests:

- Represent the graph using adjacency matrix.
- Implement function to add users (with their friendships)
- Implement function to remove users.

- Implement function to modify friendship strength (zero value means no friendship).
- Implement function to find the shortest connection path between two users. (example input: 0 4 means finding the shortest connection path between Jean and Eric)
- Implement function to find all potential friends at level n (considering 1st level is "direct friends", 2nd level is "direct friends of my friends", 3rd level is "direct friends of the friends of my friends", etc..).
- Implement function to explore all possible friendships from a given user.
- Implement function to explore whether there is a potential path between two users.
- Use Dijkstra's Algorithm to find the weakest path of friendships (or potential friendships) between two users.
- Use A* Algorithm to find the weakest path of friendships (or potential friendships) between two users using as heuristic function $h(x)$: $((\text{number of users} - 1) - (\text{number of friends of user } x))$.
- Find a way to separate users into groups of connected users. (in the example above, (Jean, Alice, David and Eric) are in the same group, (Gregory and Sandra) are in a different group)
- Get users ordered depending on their number of friends (descending order).

Notes:

- More advanced/intelligent implementation deserve higher note.
- An example input file is already provided (which is represented by the graph below), you can use it or build your own.



