Modeling a Social Network of Friendship Ratings Between Students

Nowadays, social networks have become widespread and increasingly popular. For numerous useful purposes, Social Network Analysis (SNA) exploits graph theories to study the social relationships between interacting actors. Indeed, graphs are commonly used to represent social networks in which the nodes are the social entities (e.g., individuals) and the edges describe social interactions (e.g., friendship, collaboration, trust, etc).

Figure 1: A social network

Graphs are often stored as matrices, but this method has drawbacks such as excessive memory usage and inefficiency in handling sparse graphs. Linked lists offer a more memory-efficient alternative, as they dynamically allocate memory only for existing nodes and edges, avoiding wastage common in static representations like adjacency matrices. An adjacency list represents a graph as an array of linked lists (Figure. 2). Each index in this array represents a specific node in the graph. The entry at the index i of the array contains a linked list containing the nodes that are adjacent to node i.

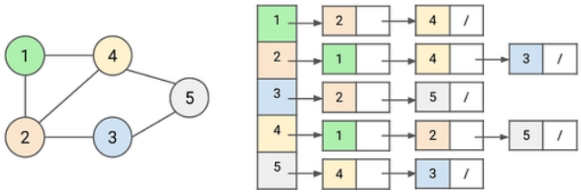

Figure 2: Adjacency list representation of graph

In this work, our aim is to analyze a real network known as Dutch college social network. It is a directed network containing friendship ratings between 32 university freshmen who mostly did not know each other before starting university. Each student was asked to rate the other student at seven different time points. A node represents a student and an edge between two students shows that the left rated the right one. The edge weights show how good their friendship is in the eye of the left node. The weight ranges from -1 for risk of getting into conflict to $+3$ for best friends. The data file is provided. It contains the necessary network information in whitespace-separated values format, with one edge per line as follows:

- First column: ID of from node
- Second column: ID of to node
- Third column: edge weight (rating)
- Fourth column: timestamp of the edge (time of recording the rating)

1 Required work

- Give the appropriate abstract data type definitions for modeling the Dutch college social network as an array of linked lists.
- Give the necessary abstract machines to manipulate the proposed structures.
- Give the operation of loading the Dutch college social network from the data file.
- Once the network is loaded, with a menu, define the following operations:
  - Add a friendship rating (edge) between two students at a specific time point.
  - Remove a student from the social network.
  - Remove a friendship rating between two students at a specific time point.
  - Find all friends of a given student at a specific time point.
  - Find shared friends of two students at a specific time point.
  - Check the strength of friendship between two students at a specific time point.
  - Find the closest friends (highest-rated friendships) of a given student.
  - Calculate the average friendship rating of a student across all time points.
- Prepare a report describing the proposed dynamic data structures and associated operations.

2 Instructions for preparing the report

The report must not exceed 12 pages and must respect the following plan:

- Cover page.
- Summary.
- Introduction: Provide background information on the topic, briefly describe the objective, and outline the organization of the report.
- Description of the proposed solution.
- List of the used abstract machines.
- List of functions and procedures used for performing the required tasks. It is preferable to give the interface of each.
- Conclusion: Summarize your work and achievements.

3 Additional information

- The deadline for submitting the work is set for 08/04/2024;
- A form will be sent to you to upload your solutions;
- Source files must be well structured, easy to read and sufficiently commented.