CSDS 293
Software Craftsmanship
2024 Fall Semester

Programming Assignment 6

Due at the beginning of your discussion session on

Oct 7 – 11, 2024

Reading

In addition to the following topics, the quiz syllabus includes any material covered in the lectures:

- In Code Complete, Section 9.3 ("Design the Routine" only) and 19.6.
- In Effective Java, Items 28, 59, 60, and 62.
- In canvas, the Pseudo-Code Cheat Sheet, the Quick Reference on Routine Names, and Repeated Code.

Grading Guidelines

In Programming Assignment 6, high-complexity and code repetition do not necessarily trigger an automatic C.

Starting with Programming Assignment 7, an automatic C (or less) will be triggered by improperly named routines.

Programming

In this assignment, you will design an algorithm for a social network analyzer. The analyzer will process a social network represented as a graph, where nodes are individuals and edges represent connections between them. Your algorithm should support the following operations:

- `addPerson(id, name)`: Add a new person to the network.
- `addConnection(id1, id2)`: Add a connection between two people.
- `removeConnection(id1, id2)`: Remove a connection between two people.
- `findInfluencers(k)`: Find the top k most influential people in the network, based on their number of connections and the influence of their connections.

- `shortestPath(id1, id2)`: Find the shortest path between two people in the network.

Your algorithm should be efficient in terms of time and space complexity, considering that the social network could potentially have millions of users.

To complete this assignment, you'll need to submit several components. Begin with an overview of your algorithm, including a narrative description and explanatory diagrams to illustrate your approach. Then, provide the pseudo-code for your algorithm, making sure to specify which classes each method belongs to and explain the abstractions your classes and methods capture.

In addition to the algorithm itself, you should include a justification for its correctness and an analysis of its running time. To demonstrate how your algorithm works in practice, include a few examples of its operation.

Remember, you're not implementing the code just yet – that will come in the next assignment. However, your pseudo-code should be detailed and clear enough that you could easily generate code from it later. After this assignment, you won't be allowed to make major changes to your pseudo-code, so be thorough in your design.

Discussion Guidelines

The class discussion will focus on the pseudo-code. The pseudo-code must be of sufficiently good quality that you can easily generate code from it in the next programming assignment. You may also be required to walk through your pseudo-code on some examples.

Submission

Submit an electronic copy of your pseudo-code and any other ancillary documents to Canvas.