Social Network Analysis for Computer Scientists
Fall 2018 — Assignment 1

`http://liacs.leidenuniv.nl/~takesfw/SNACS`

Deadline: October 1, 2018

This document contains 2 exercises that each consist of various numbered questions that together form Assignment 1 of the Social Network Analysis for Computer Scientists course taught at Leiden University.

For each question, the number of points awarded for a 100% correct answer is listed between parentheses. In total, you can obtain 100 points and 15 bonus points. Your assignment grade is computed by dividing your number of points by 10. Please do not be late with handing in your work. You have to hand in the solutions to these exercises individually. Discussing the harder questions with fellow students is allowed, but writing down identical solutions is not. Hand in your solutions, typeset using LaTeX, via the link at the course website.

For each question, clearly describe how you obtained your answer. Write down any nontrivial assumptions that you make. For the exercises that require some programming, you can use any programming language, scripting language or toolkit. All practical exercises can be done on the student workstations. In any case, always clearly describe which toolkit or programming language you used and how you obtained your answer using these tools. Include relevant source code (for example, in an Appendix).

Questions or remarks? Contact the lecturer or assistant via e-mail or ask your questions during one of the weekly lectures or lab sessions. Good luck!

Exercise 1: Neighborhoods (40p)

A directed network $G = (V, E)$ consists of a set of nodes V and a set of directed links E. For the number of nodes $|V|$ we use n, and the number of links $|E|$ will be denoted by m. The neighborhood $N(v)$ of a node $v \in V$ is defined as the set of nodes to which v links:

$$N(v) = \{w \in V : (v, w) \in E\}$$

Similarly, the reverse neighborhood $N'(v)$ can be defined as the set of nodes that link to node v:

$$N'(v) = \{u \in V : (u, v) \in E\}$$

The notion of a neighborhood can be extended by defining the neighborhood of a *set* of nodes W as:

$$N(W) = \{w \in V : v \in W \wedge (v, w) \in E\}$$

For convenience, for a node $v \in V$ we say that $N(v) = N(\{v\})$. Next, we say that the k-neighborhood $N_k(W)$ is defined as all nodes that are between 0 and k steps away from nodes in W. For the case $k = 0$ we have $N_0(W) = W$. Then for $k > 0$ we have:

$$N_k(W) = N(N_{k-1}(W)) \cup N_{k-1}(W)$$

Essentially, the k-neighborhood allows us to apply the neighborhood function to a set of nodes k times. Using these notions, it is possible to define other metrics, procedures and algorithms.

(3p) Question 1.1 Give a formal definition of the indegree and outdegree of a node using the notion of a (reversed) neighborhood.

(4p) Question 1.2 Define the reciprocity of a directed network using the notion of (reversed) neighborhoods.

(4p) Question 1.3 What do we know about a network if for all $u, v \in V$ we have: $u \in N_n(v)$?

(5p) Question 1.4 What type of network of size $n > 2$ could we be dealing with if for all $u \in V$ we have: $N_{\lfloor n/2 \rfloor}(u) = V$?

(5p) Question 1.5 Define the metric of closeness centrality using the notion of a (k-)neighborhood.

(5p) Question 1.6 Describe how to use the neighborhood function in an algorithm that checks if the network is actually a directed acyclic graph.

(6p) Question 1.7 The *periphery size* of a connected undirected network is the size of the set of nodes that is on either one of the the ends of the network's shortest paths of maximal length. Alternatively, the periphery size is the number of nodes with maximal eccentricity. Give a definition of the periphery size, using the neighborhood function.

(8p) Question 1.8 Give an algorithm that uses the neighborhood function to count the number of triangles in a connected undirected network. Use consistent pseudo-code or concise words. What is the time complexity of your algorithm?

Exercise 2: Mining An Online Social Network (60p)

This is a practical exercise, for which you can use any toolkit or programming language. The social network datasets `medium.tsv` and `large.tsv` (tab-separated and with UNIX line endings) can be found in the Leiden ULCN environment in the shared UNIX folder: `/vol/share/groups/liacs/scratch/SNACS` For convenience and speed, you should copy the files to the local hard disk `/scratch/` of the machine you are working on. Each file contains a list of friendships of an online social network of the form `userA[whitespace]userB[newline]` and represents one directed link from a person identified by `userA` to a person identified by `userB`. You may assume that these identifiers are integers that fit in a 4-byte `signed int` in C++. Processing the file `medium.tsv` should be possible on a student workstation with 16GB memory, using for example GEPHI. A larger online social network is given in the file `large.tsv`, wich will likely not be processable using standard toolkits such as GEPHI, requiring the use of for example NETWORKX or another CLI package.

Answer each the following six questions for **both** of the datasets `medium.tsv` and `large.tsv` (hence, points are also given 2×), and remember to write down how you obtained your answer and to include (pointers to) relevant source code.

(2×2p) Question 2.1 How many directed links does this network have?

(2×2p) Question 2.2 How many users (nodes) does this social network have? Hint: a node counts as a node if it is a source or a target of a link.

(2×5p) Question 2.3 Give the indegree and outdegree distribution of this network (plot for each degree value how often it occurs). Present the results in a representative diagram with proper axes (and labels), for example generated using a simple tool such as GNUPLOT or MATPLOTLIB.

(2×4p) Question 2.4 How many weakly connected components and how many strongly connected components does this network have? How many nodes and links are in the largest strongly connected component of this network?

(2×2p) Question 2.5 Give the exact or approximated average clustering coefficient of this network.

(2×5p) Question 2.6 Give the exact or approximated distance distribution of the largest weakly connected component of this network as a diagram.

(20p) Question 2.7 Visualize the social network in `medium.tsv` as a vector graphic so that it can be printed on A4 paper. Give the size and optionally the color of a node a sensible meaning based on node centrality, and explain your choices. State which visualization algorithm you used and how you choose its parameters. Include your network as a full-page A4 vector graphic PDF in your report, or include it as a separate file.

(15p, bonus) Question 2.8 The file `huge.tsv` contains over 5 million nodes and over 1 billion edges. Answer questions 2.1 through 2.6 above. You will need to use approximation and.or a more advanced software package and environment (e.g., GRAPH-TOOL), or write efficient code yourself.