Social Network Analysis for Computer Scientists
Fall 2025 — Assignment 1

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

Deadline: September 29, 2025

This document contains two 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 10 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.

Clearly and concisely describe how you obtained each answer. Write down any nontrivial assumptions that you make. For the exercises that require programming, you can use any programming language, scripting language or toolkit; a trivial option is to use Gephi and NetworkX that were covered in the course. In any case, always clearly describe which tools and languages you used and how you obtained your answer using these tools. Recall that omission of a reference to source material is considered plagiarism. When asked for an algorithm, use simple and consistent pseudo-code. Include relevant source code in an Appendix that you reference in your answers. Please use the `listings` package for including source code. If you used an interactive notebook, use `nbconvert` to convert the notebook to regular source code.

Submission. Hand in your solutions via Brightspace, in one `.pdf` file, typeset using LaTeX. Remember to specify your name and student ID (ULCN number) on top of your assignment.
Do not copy the full text of each question into your document (if you do, you will be asked to resubmit). Just stating the question number and then your answer, is sufficient. If you really need to submit multiple files, please attach them all in one submission. If you want to make a new submission, replacing your previous submission, make sure to again include all the files in that submission. Thank you for taking this into consideration.

Questions or remarks? Ask your questions during one of the weekly lectures or lab sessions, or send an e-mail. 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 it for a *set* of nodes W as:

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

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 measures, procedures and algorithms. Throughout this exercise, we consider the scenario in which we can iterate over all nodes in V and apply the (reversed) neighborhood function, but cannot iterate over all edges in E.

(2p) Question 1.1 Use formal notation to define the outdegree and indegree of a node $v \in V$, making use of the (reversed) neighborhood function.

(2p) Question 1.2 In a directed network, the *combined degree* of a node v is the number of unique neighbors connected to that node through either an incoming or an outgoing link. Give a formal definition of the combined degree using the notion of a (reversed) neighborhood.

(4p) Question 1.3 Give a formal definition using the $\forall$ quantifier and the (reversed) neighborhood function that holds true if for all nodes $v \in V$, at least half of its incident edges are *reciprocated*.

(3p) Question 1.4 Define a node v's *reversed k-neighborhood* by combining the concepts of k-neighborhood and reversed neighborhood, and briefly explain what it measures.

(5p) Question 1.5 Give a formal definition or pseudocode of an algorithm that holds or returns true if two given nodes $u, v \in V$ are in the same *strongly connected component* of a given network, using the notion of a (reversed) (k-)neighborhood.

(6p) Question 1.6 Give a formal definition or pseudocode of an algorithm to compute the number of unique *directed triangles* involving a given node $v \in V$, making use of the (reversed) (k-)neighborhood function.

Assume from now on that the network has a symmetric edge set, modeling that it is undirected. Also assume there is one connected component.

(5p) Question 1.7 Give a formal definition or pseudocode of an algorithm to compute the network *center* using the notion of a (k-)neighborhood. Hint: the center is set of nodes with the minimal eccentricity value over all nodes.

(2p) Question 1.8 A node u is said to *dominate* a node v if all neighbors of v are also neighbors of u. Give a formal definition of this dominance relation, making use of the neighborhood function.

(8p) Question 1.9 Give pseudocode of an algorithm that identifies the set of all non-dominated nodes. What is the complexity of your algorithm?

(3p) Question 1.10 Give pseucode of an algorithm that identifies the set of all non-dominated nodes, but now without access to the full set V, but only with initial access to an arbitrary node $v \in V$ and the neighborhood function.

Exercise 2: Mining An Online Social Network (60p + 10p bonus)

This is a practical exercise. Two social network datasets can, after filling in your ULCN ID, be obtained through: `https://liacs.leidenuniv.nl/~takesfw/SNACS/a1data.php`
It should give you a `medium.tsv` and `large.tsv` file. Each file contains a list of social network friendships in edge list format, so of the form
`userA[tab]userB[newline]`
A line thus represents one directed link from a person identified by `userA` to a person identified by `userB`. Assume that these identifiers are integers that fit a 4-byte `signed int` in C++.

Questions about visualization can likely best be done using GEPHI, whereas computing measures and distributions requires the use of programming and packages such as NETWORKX.

Answer each of the following seven questions for `medium.tsv` and `large.tsv` (hence, up to Question 2.7, points are also given 2×). Remember to write down what assumptions and choices you made in obtaining your answer. Present the numbers asked in Question 2.1, 2.2, 2.4, 2.5 and 2.7 in a table, with a row per requested value, and a column for each dataset. You should still list each question and detail how you obtained your answer, through a clear reference to (lines of) relevant Appendix source code. Display your referenced diagrams with neat, properly scaled, readable, labeled axes and captions; the latter also holds for tables. Plots can for example be generated using PYPLOT/MATPLOTLIB.

(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 user counts as a node if it is a source or target of at least one link.

(2×4p) Question 2.3 Give the indegree and outdegree distributions of this network using a proper diagram.

(2×3p) Question 2.4 How many weakly connected components and strongly connected components are there? How many nodes and links do the largest strongly and largest weakly connected component have? (6 answers per network)

(2×3p) Question 2.5 Give the (exact or approximated) average clustering coefficient of this network. Explain how you account for directionality.

(2×6p) Question 2.6 Give the (exact or approximated) distance distribution of the largest weakly connected component of this network as a diagram.

(2×2p) Question 2.7 What is the (exact or approximated) average distance between any two node pairs in the largest weakly connected component of this network?

(16p) Question 2.8 Visualize the social network in `medium.tsv`, for example using GEPHI. Give the size and the color of a node a sensible meaning based on node centrality, and elaborate on your choices. State which visualization algorithm you used and how you have chosen its parameters. Include your visualization as full-page vector graphic figure in your report.

(10p, bonus) Question 2.8 The "huge" dataset contains over 5 million nodes and 800 million edges:
`/vol/share/groups/liacs/scratch/SNACS/huge.tsv`
`/data/SNACS/huge.tsv`
The files are identical, where the first is in the university-provided (remote) Linux environment and the second in the LIACS data science lab environment (see `https://liacs.leidenuniv.nl/ict` or `https://rel.liacs.nl`, respectively, for access instructions).
Answer Questions 2.1 through 2.7 above for this dataset. You may need to make more use of approximation and/or a more advanced software package and environment (e.g., you should investigate GRAPH-TOOL, IGRAPH or SNAP), or write efficient code yourself.