Social Network Analysis for Computer Scientists
Fall 2025 — Assignment 2

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

Deadline: October 27, 2025

This document contains various numbered questions that together form Assignment 2 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. **When asked for an algorithm, use simple and consistent *math-style* pseudo-code.** For the exercises that require programming, you can use any programming language, scripting language or toolkit; a trivial option is to use NetworkX as 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.

Include relevant source code in an Appendix that you reference in your answers. Please use the `listings` package for including source code. Consider referencing relevant lines in your source code as a way to indicate how you obtained your answer. If you used an interactive notebook, use `nbconvert` to convert the notebook to regular source code, which you may have to make "readable" by adding comments in the appropriate places.

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: Advanced concepts, projection and centrality (18p)

The *average clustering coefficient* $C(G)$ of a connected undirected network $G = (V, E)$ indicates the extent to which nodes cluster together, and is based on the *node clustering coefficient* $c(v)$ of the nodes $v \in V$. The two are defined as follows.

$$C(G) = \frac{1}{n} \cdot \sum_{v \in V} c(v)$$

$$c(v) = \frac{2 \cdot |\{(u, w) \in E : (u, v) \in E \wedge (v, w) \in E\}|}{deg(v) * (deg(v) - 1)}$$

Here, $deg(v)$ is the degree of node v, and to avoid divide by zero errors, $c(v)$ is assumed to be equal to 0 if $deg(v) \leq 1$. Assume that $n = |V|$. The undirected graph is modelled as a directed graph with a symmetric edge set.

(3p) Question 1.1 Name two types of networks that have an average clustering coefficient of 0 by definition and one type of network that has an average clustering coefficient of 1 by definition.

(4p) Question 1.2 Prove that for any undirected network $G = (V, E)$, there exists a node $v \in V$ such that its node clustering coefficient value is greater than or equal to the average local clustering coefficient of the entire network.

(4p) Question 1.3 Now assume that we have an undirected graph $G = (V_L, E)$, which is a projection of a bipartite graph $G' = (V_L, V_R, E')$ (as defined in the Lecture 3 course slides). If the connected component size distribution of G equals the degree distribution of V_R, then what is known about the degrees of the nodes in V_L in G' ?

(4p) Question 1.4 Consider again a *bipartite graph*, with its two types of nodes, and suppose there are n_1 nodes of type one and n_2 nodes of type two. Show that the mean degrees k_1 and k_2 of the two node types are given by

$$k_2 = \frac{n_1}{n_2} \cdot k_1$$

(3p) Question 1.5 Centrality measures are commonly used to assess the importance of individual nodes, based on their structural position in the network. By sorting nodes in the network by their centrality value, we can create a ranking of nodes. Discuss two special types of networks that we could be dealing with if the node ranking produced by *degree centrality* is equal to that of *betweenness centrality.*

Exercise 2: Diameter computation (12p)

Apply the BoundingDiameters algorithm on paper (i.e., by hand) to find the exact diameter (maximum distance, length of a longest shortest path) of the undirected graph in Figure 1. The algorithm is discussed during the lectures (see, e.g., `http://liacs.leidenuniv.nl/~takesfw/pdf/diameter.pdf`) and explained in detail in [1].

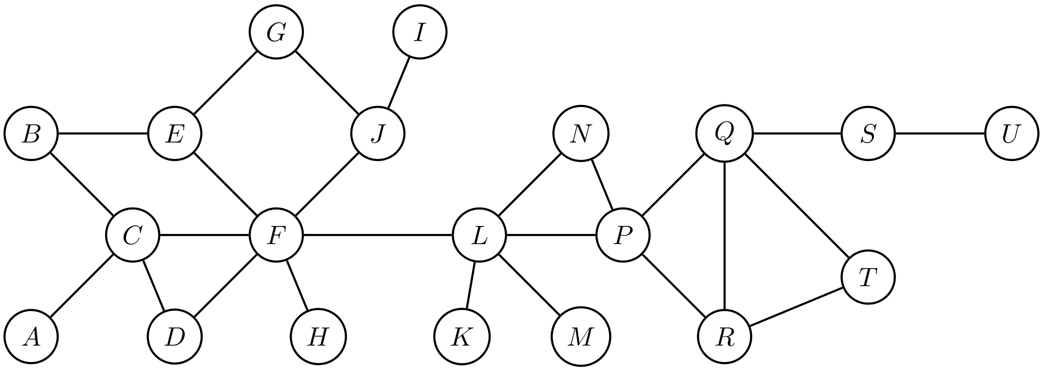

Figure 1: An undirected graph with 20 nodes.

You do not have to do the prepruning step discussed in the paper. **Explain your steps in detail**, and mention any nontrivial assumptions. As a selection strategy, alternate between choosing the node with the largest upper bound value and the node with the smallest lower bound value, breaking ties by taking the node with the highest degree. How many iterations did it take to compute the diameter, and how does this compare to the naive method for diameter computation?

[1] F.W. Takes and W.A. Kosters, Determining the Diameter of Small World Networks, in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 1191-1196, 2011. doi: `http://dx.doi.org/10.1145/2063576.2063748`

Exercise 3: Twitter network analysis (60p+10p)

This is a practical exercise, for which you can use any toolkit or programming language. Two samples of Twitter datasets can be found at

`https://liacs.leidenuniv.nl/~takesfw/SNACS/twitter-small.tsv` and
`https://liacs.leidenuniv.nl/~takesfw/SNACS/twitter-larger.tsv`.

The full dataset (not required until bonus Question 3.7) can be found at

`/vol/share/groups/liacs/scratch/SNACS/twitter.tsv` and
`/data/SNACS/twitter.tsv`.

The above two files are identical, where the first is in the university/ISSC-provided remote Linux environment (see their helpdesk portal `https://liacs.leidenuniv.nl/ict`) and the second in the LIACS data science lab environment (see `https://rel.liacs.nl`, only accessible internally).

Up until now, we have only looked at social network data which was already in a nicely formatted edge list. In practical network analysis research, this rarely ever happens. Therefore we will now work with real raw data from a Twitter crawl [2]. The dataset `twitter.tsv` contains over $450,000,000$ tweets, crawled from June 2009 to December 2009. The file `twitter-small.tsv` contains a small subset of these tweets that can, after preprocesing, be handled with Gephi, whereas `twitter-large.tsv` contains a bit larger subset that can be analyzed using for example NetworkX. Each line of these files contains one tweet, consisting of three tab-separated (‘`\t`’) fields denoting the timestamp, user who sent the tweet and the content of the tweet. For example:

```
2009-07-05 14:07:18    aeneas    Hi @achilles, how are you? #old
```

In the tweet content, a word starting with the `@` symbol (such as `@achilles`) means that user `achilles` is being mentioned by user `aeneas`, indicating that the tweet by `aeneas` was directed at or specifically about `achilles`. We refer to this as a *mention*. Mentions are the most direct sign of public communication on Twitter. Tweets can also be directed at more than one user.

The *mention graph* is a Twitter network represented by a directed graph. The set of nodes consists of users (anyone sending out a tweet or being mentioned by someone else in a tweet). The set of links consists of all user pairs (x, y) such that user x mentioned user y at least once. The mention graph is in fact a *weighted* directed graph where the number of times a user x mentions another user y is the link weight.
Now, for the `twitter-small.tsv` dataset, answer Question 3.1–3.6.

(16p) Question 3.1 Extract the mention graph from the Twitter data. Relevant steps to do this could be:

- Parse the input file line by line (for example using Python or Perl).
- Generate the adjacency list: for each user (identified by its username), keep a list of the users that this user mentions, and count the number of mentions.
- Output the adjacency list as a weighted edge list `csv`-file.

Discuss the steps that you took, and describe the issues that you ran into while parsing this “real-world” data, and how you solved them. For example, discuss possible text mining and parsing issues. Carefully think of how capture a valid Twitter username. Important: from your description (in words; do not just give the code), it should be possible to unambiguously reproduce your network dataset; points are mostly awarded for reproducibility of your approach from your description, and not merely for code correctness.

(12p) Question 3.2 Present relevant statistics of your mention graph in a table, including at least

- (1p) the number of nodes and edges,
- (2p) number and size of the strongly and weakly connected components,
- (1p) density,
- (1p) (approximated) average node clustering coefficient, and
- (1p) (undirected) (approximated) average distance in the giant component.

Moreover, generate the following diagrams and include them in your report as figures:

- (3p) indegree and outdegree distributions, and
- (3p) (undirected) (approximated) distance distribution of the giant component.

(8p) Question 3.3 Determine the top 20 users based on three different centrality measures (for example, betweenness, closeness and degree centrality). Mention how you deal with directionality. Discuss the results. Think of a way to compare the similarity of the rankings using some measure, apply it, and briefly interpret the results.

(6p) Question 3.4 Apply a community detection algorithm to the giant component of your mention graph, and manually interpret and discuss the results.

(6p) Question 3.5 A weight distribution indicates how often each link weight occurs. Present this distribution using a proper diagram. What type of distribution does this appear to be?

(12p) Question 3.6 Answer Question 3.2 for the larger dataset `twitter-larger.tsv`.

(10p, bonus) Question 3.7 Answer Question 3.2 for the full 450M tweet dataset given in the file `twitter.tsv`. This is very challenging, and may require you to either use large computing sources, or to be smart/selective in your data processing steps, for example by systematically filtering certain users and/or links, for example based on some threshold for the number of mentions. Only do this after answering all the other questions, and remember to explain your approach to dealing with the sheer size of this dataset.

[2] J. Yang and J. Leskovec, Temporal variation in online media, in *Proceedings of WSDM*, pp. 177–186, 2011. Available at `dx.doi.org/10.1145/1935826.1935863`