Social Network Analysis for Computer Scientists
Fall 2024 — Assignment 2

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

Deadline: October 28, 2024

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.

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: Diameter computation (10p)

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 and explained in:
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` or see
`http://liacs.leidenuniv.nl/~takesfw/pdf/diameter.pdf`.

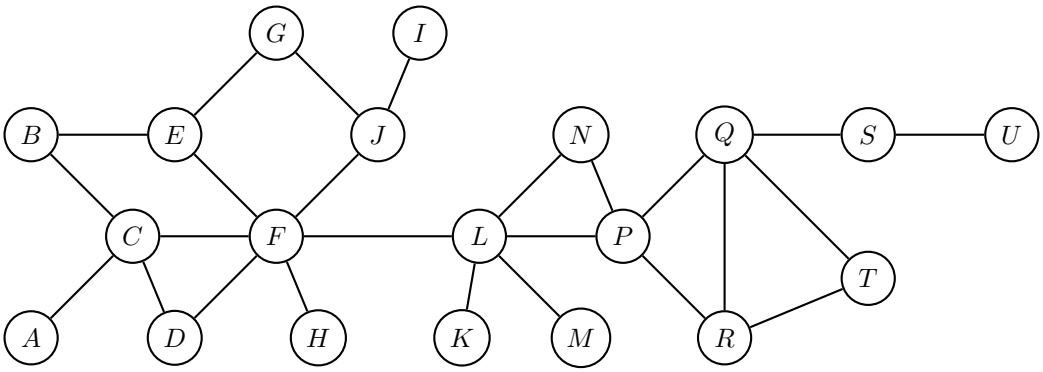

Figure 1: An undirected graph with 20 nodes.

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?

Exercise 2: Museum as a Network (30p)

We want to model the layout of Amsterdam's Rijkmuseum as a network in an attempt to better understand the museum's structure. Figure 2 shows a (simplified) floorplan of the museum. In the corresponding network dataset, edges represent the connections between rooms in the museum. Two rooms are considered connected if there is a doorway between them, or if they both contain a stair symbol *with the same name*. Hint: the edges of room 0.1 are $\{\{0.1, 0.2\}, \{0.1, 0.3\}, \{0.1, 1.15\}, \{0.1, gthall\}, \{0.1, 3.1\}\}$. In this exercise you can make use of tooling such as NetworkX.

(2p) Question 2.1 What type of graph are we dealing with when we consider a graph made out of a two-dimensional (so, one floor) physical lay-out?

(8p) Question 2.2 Create the network dataset and visualize the resulting network, with room names visible as node labels. Use `nx.draw(g, with_labels = True)` in NetworkX, and include this image in your report. State how many nodes and how many edges your graph has.

Question 2.3 A regular graph is a graph in which each node has the same degree. A boring (and theoretically impossible) grid-shaped floorplan of a museum would translate to an infinite regular graph with degree 4, in which each room links to the room above, below, to the left and to the right of it.

(2p) Question 2.3a What do we know about the clustering coefficient of such a graph?

(2p) Question 2.3b Can you name a type of graph in which the clustering coefficient behaves in the same way, but is not a regular or empty graph?

Question 2.4 We want to visit every room in the museum, and seek to find a path that realizes this objective with a minimal number of duplicate room visits.

(3p) Question 2.4a Explain how to compute such a path, and give this path for your constructed network. You can use a toolkit such as NetworkX.

(3p) Question 2.4b Does the resulting path seem reasonable? Would you follow it for an actual museum visit? Can you suggest a change to the network model that could improve the construction of such a path?

(5p) Question 2.5 We are not interested in seeing paintings anymore. Instead we would like to walk through all the doorways and stairs in the museum. You plan to apply your approach from Question 2.2 again. You have, however, lost your floorplan, and have only the network dataset left. How can you transform this previously constructed network of the museum in order to create a network suitable for finding the solution to this new goal of visiting all doorways and stairs? Give either a description of this transformation using formal notation, or an algorithm for constructing the new network out of the old one.

(5p) Question 2.6 The museum wants to install a first-aid kit in one of the rooms, such that in case of an emergency (which could occur in any room), it will be as close as possible to the place of the emergency, in terms of how many rooms should be traversed to get there. Explain how to use the network structure to find the most suitable room. List the three most suitable rooms.

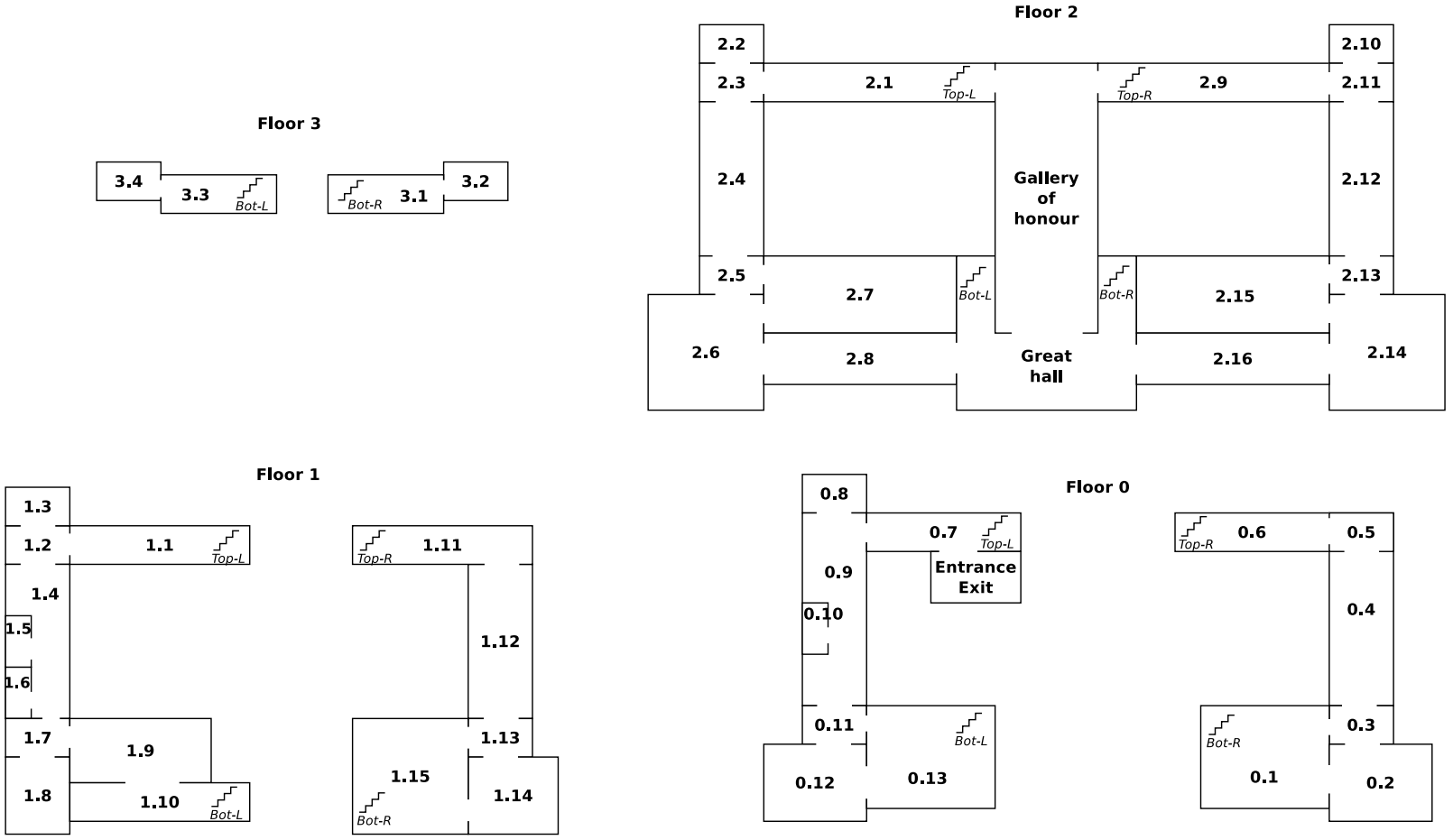

Figure 2: A (simplified) flooarplan of Rijkmuseum in Amsterdam. Note: rooms 2.7 and 2.15 do not have a door leading to the Great Hall.

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, and not merely for 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.

(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 systematically filter certain users and links, for example based on some threshold for the number of mentions. If you succeed on $x\%$ of the data, you can get up to $x\%$ of the 10 bonus points. Only do this after answering all the other questions, and remember to explain your steps.

[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`