Complex and Adaptive Dynamical Systems

WiSe 23/24 | 23.10.2023

C. Gros, D. Nevermann

Exercise Sheet #1

Deadline: 30.10.2023, 12:00

How to submit your solutions

(a) In order to earn the points required to pass the course you need to submit a clear and concise solution, documenting your thought process, to the problems given throughout the course before the respective deadlines.

(b) While this is not mandatory, we recommend that you write out your solutions in LaTeX and submit them as a PDF.

(c) For each problem, you are required to provide all relevant files, named in the format
<YourName>_problem<number>_<additionalInfo>.xyz.
Please do not use spaces in file names.

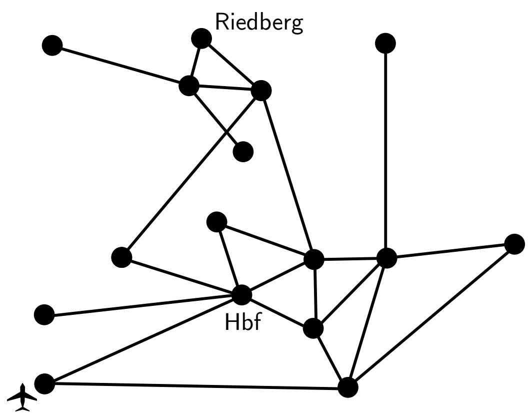
(d) To hand in your solution to a sheet, send an e-mail to your tutor
lkiefer@itp.uni-frankfurt.de
containing the bundled files from (c) to all problems named in the format
<YourName>_sheet<number>.zip.

(e) Grades for a sheet are given on a scale of 0 to 20 points. If there are more than 20 points achievable on a given sheet, 20 is still the maximal grade.

(f) The **deadline for submission** for a given sheet will always be indicated at the top of the sheet.

Problem 1 (*Properties of Graphs*) (10 points)

The graph given below shows an abstraction of a part of the Frankfurt public transport network.



Evaluate the following properties of graphs:

(a) coordination number z,

(b) connection probability p,

(c) network diameter l,

(d) and clustering coefficient C.

Compare the network diameter and clustering coefficient with the values you would expect for a random graph with the same coordination number.

Problem 2 (*Generating Erdös–Rényi Graphs*) (10 points)

In this problem we want to implement a generator for an Erdös–Rényi in Python.

(a) Write a function in Python that takes a number of nodes N and a connection probability p as input. You can label the nodes with numbers $0, \ldots, N-1$. The function should generate the edges for an Erdös–Rényi graph and return them as list of edges, where an edge between nodes i and j should be represented as a tuple (i, j). Generate and print out the list of edges for a graph with $N = 100$ and $p = 0.01$. You may use the code snippet below as guidance.

Hint: You can use the `itertools` module to generate a list of all possible edges. For that use `itertools.combinations(range(N), 2)`. Furthermore, using the `random` module, you can generate a random number between 0 and 1 using `random.random()`.

```python
import itertools
import random

def generate_erdos_renyi(N, p):
    # Write your code here.
    return edge_list

if __name__ == "__main__":
    N = 100
    p = 0.01
    print(f"Generating an Erdos-Renyi Graph with N = {N}
        nodes, p = {p}. Edges are:")
    print(generate_erdos_renyi(N, p))
```

(b) To test your implementation, generate a large number (e.g. 10 000) of Erdös–Rényi graphs with $N = 100$ and $p = 0.01$ and compute the average number of edges. From the theory, what is the number of edges you would expect to see approximately?