

7. Übungsblatt
Diskrete Strukturen (Winter 2019/20)

Bauhaus-Universität Weimar, Professur für Mediensicherheit

Dr. Stefan Lucks, Jannis Bossert

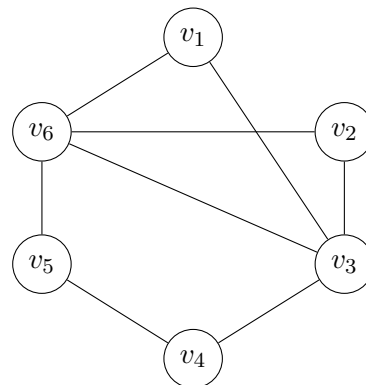
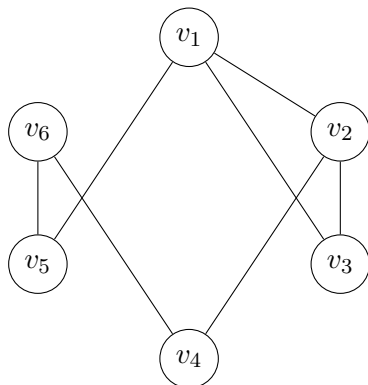
URL: <http://www.uni-weimar.de/de/medien/professuren/mediensicherheit/teaching>

Abgabe: Bis zum 4. Februar 2020 vor Beginn der Übung oder per E-Mail an jannis.bossert@uni-weimar.de. Lösungen sind bevorzugt in LaTeX zu verfassen. Ein Template finden Sie auf der Übungsseite der Veranstaltung.

Aufgabe 1 – Suche (2+2 Punkte)

Gegeben seien die folgenden einfachen ungerichteten Graphen.

- a) Geben Sie für die abgebildeten Graphen je einen Eulerkreis und einen Hamiltonkreis an oder begründen Sie jeweils nachvollziehbar warum es keinen geben kann.
- b) Geben Sie für den rechten Graphen jede mögliche Knotenfolge für eine Tiefensuche mit Startknoten v_1 an (die Suche soll alle Knoten genau einmal durchlaufen).



Aufgabe 2 – Eulersche Polyederformel (3 Punkte)

Beweisen Sie Satz 120 aus der Vorlesung.

Satz 120:

Für jeden zusammenhängenden planaren Graphen (E, V) mit $|V| \geq 1$ Knoten $|E|$ Kanten und a Gebieten gilt:

$$a + |V| = |E| + 2$$

Hinweis: Induktion nach $|E|$.

Aufgabe 3 – Graphentheorie (3+3 Punkte)

- a) Zeigen oder widerlegen Sie: Für jeden einfachen ungerichteten Graphen $G = (V, E)$ mit $|V| \geq 2$ existieren zwei Knoten $v, v' \in V$ mit $v \neq v'$ und $\deg(v) = \deg(v')$.
- b) Für einen Knoten v in einem gerichteten Graphen sei $\deg_{\text{in}}(v)$ die Anzahl der eingehenden Kanten und $\deg_{\text{out}}(v)$ die Anzahl der ausgehenden Kanten. Zeigen oder widerlegen Sie: Ein einfacher gerichteter Graph $G = (V, E)$ ohne isolierten Knoten hat einen Eulerkreis $\implies G$ ist zusammenhängend und für alle $v \in V$ gilt $\deg_{\text{in}}(v) = \deg_{\text{out}}(v)$.

Aufgabe 4 – PageRank (4 Punkte)

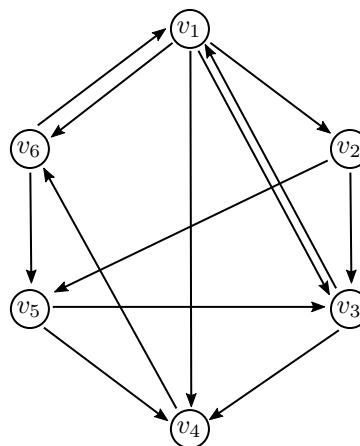
Studieren Sie Kapitel 7.4 der Vorlesung (s. Folien 312ff.). Gegeben sei der unten abgebildete gerichtete Graph aus Webseiten $V = \{v_1, \dots, v_6\}$. Die gerichtete Kante (v_i, v_j) beschreibt, dass ein Nutzer von Webseite v_i zu Webseite v_j gelangen kann. Die Wahrscheinlichkeit dass Nutzer einem Link folgen ist für alle Links einer Webseite gleich. Sei der Dämpfungsfaktor d die Wahrscheinlichkeit mit der ein Nutzer zu keiner der verlinkten Webseiten geht. Schreiben Sie ein Pythonscript das für einen gegebenen Graphen und für n Iterationen die PageRanks der Webseiten ermittelt.

Beispielaufruf:

```
$ python3 pagerank_<xxxxxxx>.py -i graph.txt -n 10 -d 0.2
0.297 0.098 0.241 0.304 0.234 0.363
```

Das Dateiformat des Graphs, hier in **graph.txt**, soll wie folgt sein: Die erste Zeile “**p 6 13**” im folgenden Beispiel sagt aus dass der Graph 6 Knoten und 13 Kanten hat; die darauffolgenden Zeilen beschreiben je eine Kante, z. B. “**e 1 2**” die gerichtete Kante von Knoten v_1 zu v_2 :

```
p 6 13
e 1 2
e 1 3
e 1 4
e 1 6
e 2 3
e 2 5
e 3 1
e 3 4
e 4 6
e 5 3
e 5 4
e 6 1
e 6 5
```



Schicken Sie Ihre Lösung als Anhang einer E-Mail als Pythondatei

pagerank_<IhreMatrikelnummer>.py

an jannis.bossert@uni-weimar.de mit dem Betreff [DS WS2019/20 Beleg 7]. Es reicht für **IhreMatrikelnummer** eine Matrikelnummer Ihrer Gruppe. Die vollständigen Namen und Matrikelnummern sollen als Kommentar in der Pythonabgabe stehen.