Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Lehrstuhl für
INFORMATIK I
Algorithmen & Komplexität

Institut für Informatik

Algorithmische Graphentheorie

Sommersemester 2021

1. Vorlesung

Graphen: Eine Einführung

Lehrstuhl für Informatik I

Alexander Wolff
Johannes Zink

Termine & das Kleingedruckte

Vorlesung per Video:

– Folien und Videos auf WueCampus
– Mittwochs, 10:30–11:30, Fragestunde zum Vorlesungsvideo per Zoom und grundsätzlich auch schriftlich per uw-Chat.
– Dozent: Prof. Alexander Wolff (Büro 01.001, Gebäude M4, Sprechstunde: Mi, 13:30–14:30)

Alle Links dazu auf WueCampus!

(Termin bitte per Email abmachen, dann Zoom o.ä.)

Übungen:

– Organisation: Johannes Zink (Büro 01.007, Gebäude M4, per Email erreichbar)
– Tutoren: Diana Sieper, Vasil Alistarov, Tim Gerlach, Samuel Wolf
– Freitags, 8:30 (Gruppe 1), 10:15 (Gruppe 2 & 3), 12:15 (Gruppe 4)
– Erstmals schon diese Woche, **16.4.!**

Termine & das Kleingedruckte

Übungsaufgaben:

– Bearbeitung in Gruppen von max. je zwei Teilnehmern
– Ausgabe: mittwochs via WueCampus
– Abgabe: dienstags, 12:00 Uhr, auf WueCampus (nur pdf)
Bitte möglichst mit LaTeX o.ä. schreiben!

Übungsmodus:

– Kein „klassischer“ Übungsbetrieb mit der ganzen Übungsgruppe
Ausnahme: 1. Übung am 16.4.
– 15-Min.-Slots in Übung am 16.4. mit Ihrem Übungsleiter ausmachen
– Treffen über Zoom
– Individuelle Besprechung, Fragen zum letzten Übungsblatt, Fragen/Diskussion/Bearbeitung des aktuellen Übungsblatts
Machen Sie sich vorab Gedanken, was Sie besprechen wollen!
– Zusätzliche Plattform: RocketChat der Uni Würzburg (uw-Chat)

2x Anmelden!

Bitte melden Sie sich *sofort* bei WueStudy und WueCampus an

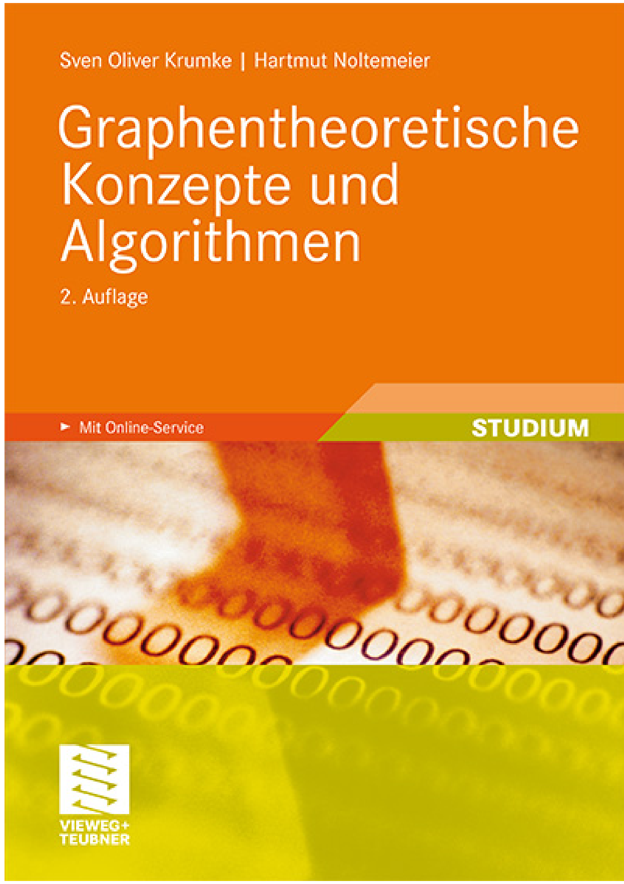
- Übungseinteilung erfolgt über WueStudy
  (Die Übungszeiten sind wichtig für die persönliche Besprechung Ihrer Kleingruppe mit dem Übungsleiter.)
- WueCampus für Kommunikation, z.B. falls eine Übungsaufgabe fehlerhaft oder missverständlich ist.
  (Zum Einschreiben klicken Sie auf das kleine Zahnrad oben links und wählen dann “Mich in diesen Kurs einschreiben”.)

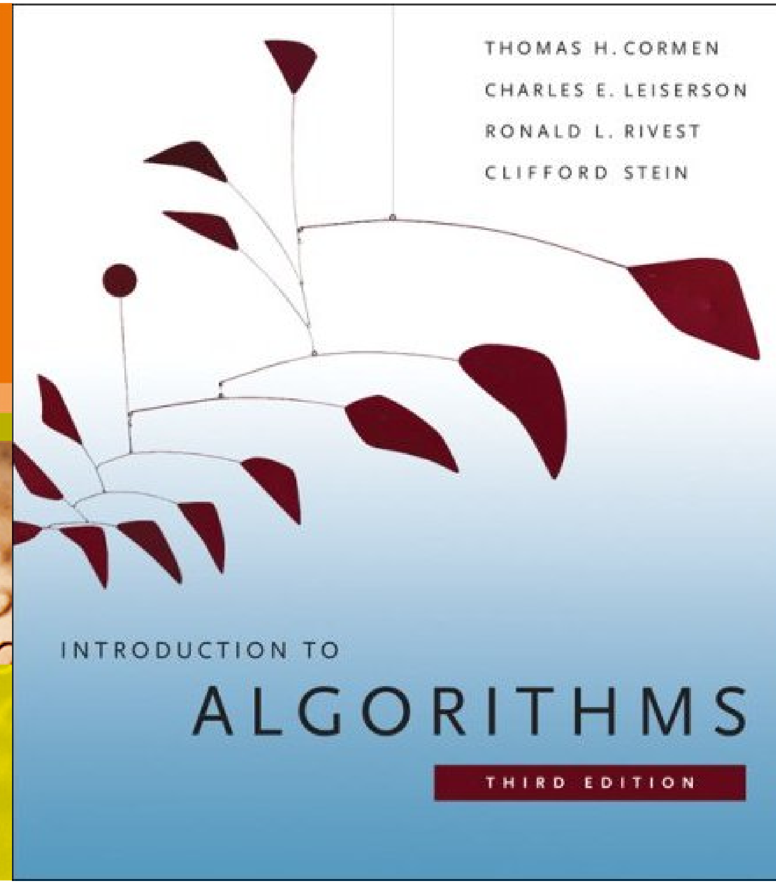
Klausuren:

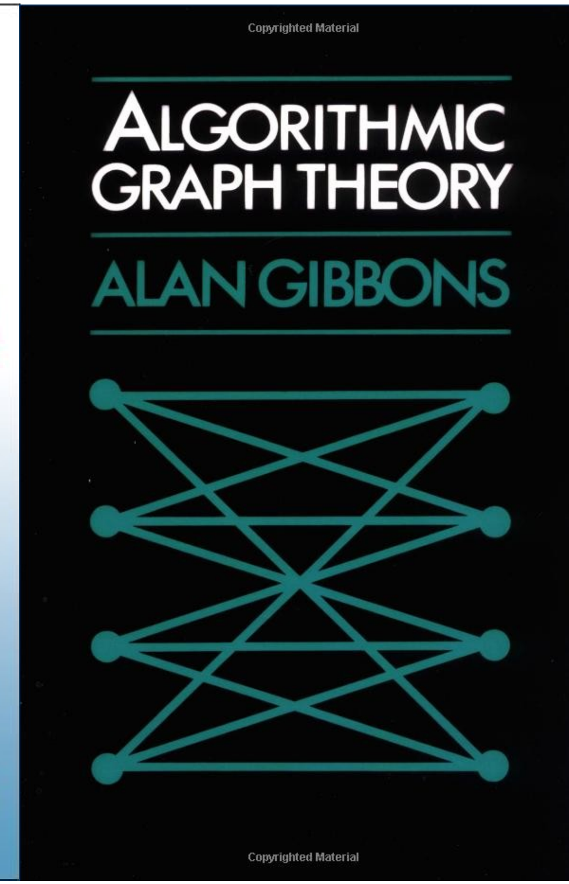
– 1. Termin: 26.07.2021, 10:00–12:00 Uhr, Z6 [Anm. bis 15.07.]
– 2. Termin: Anfang Oktober 2021

- Wenn Sie sich nicht fristgerecht bei WueStudy anmelden, ist es für uns **unmöglich** Ihre Note zu verbuchen.

Bücher zur Vorlesung







[KN] [CLRS] [G]

Hauptreferenz; elektronische Kopie über die Unibib erhältlich

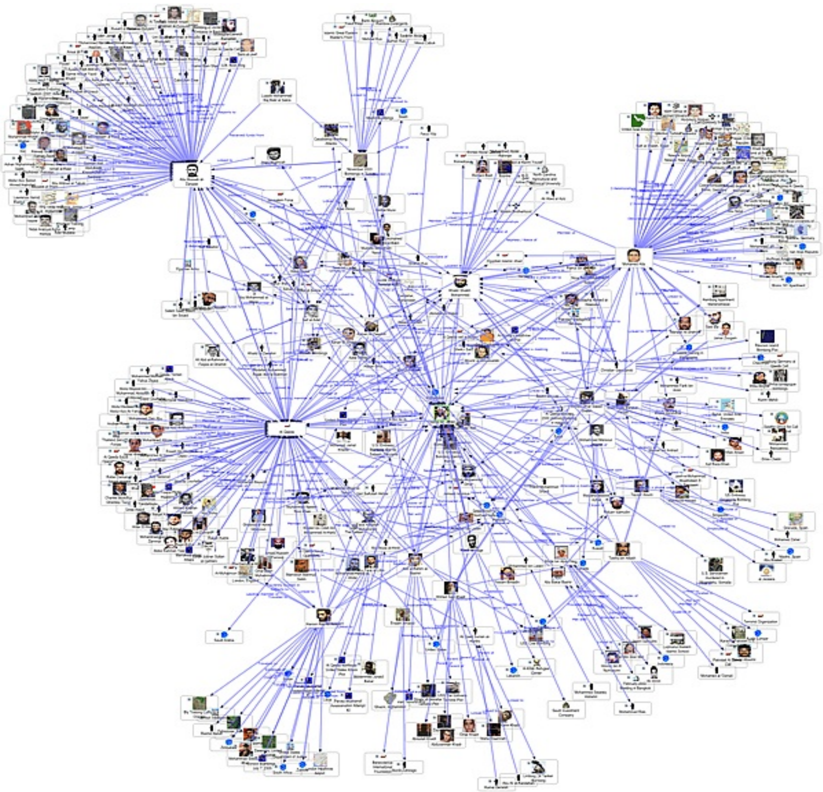
Voraussetzungen

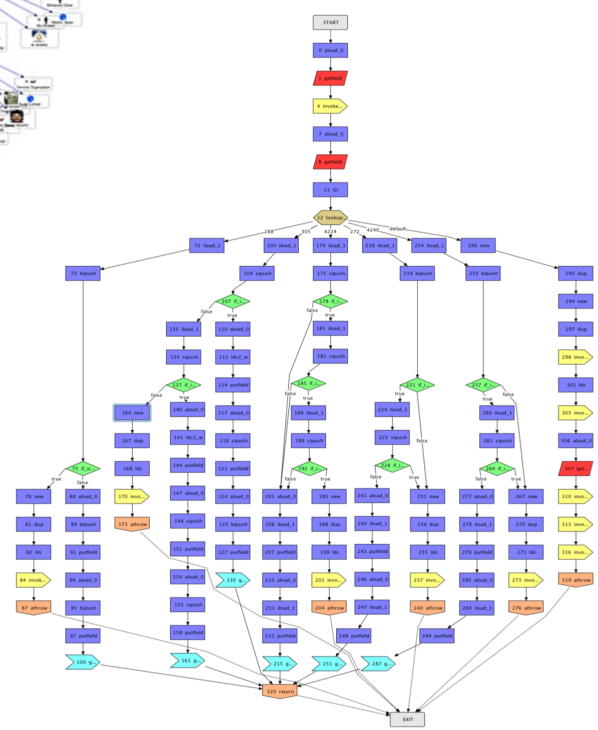
Wissen aus der Vorlesung *Algorithmen und Datenstrukturen*:

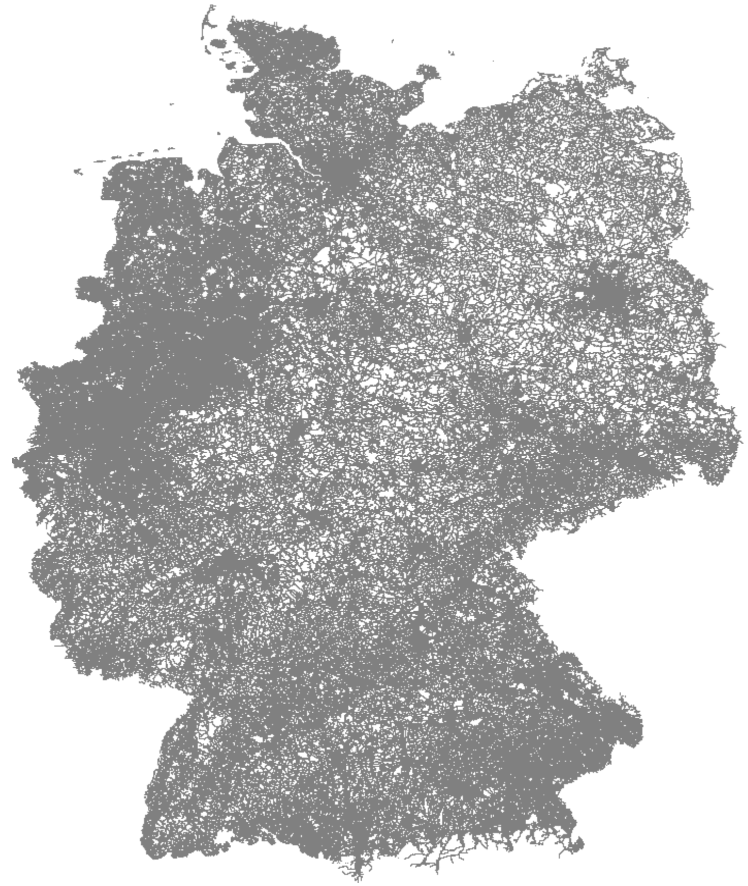
- Graphdurchlauf-Strategien
  - Breitensuche
  - Tiefensuche
- Berechnung kürzester Wege
  - Breitensuche
  - Algorithmus von Dijkstra
- Minimale Spannbäume
  - Algorithmus von Kruskal
  - Algorithmus von Jarník–Prim

Repetitorium in der allerersten Übung

Graphen



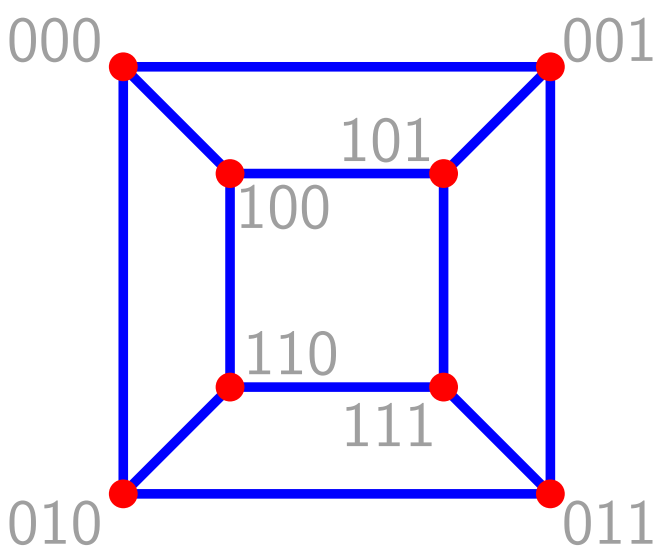


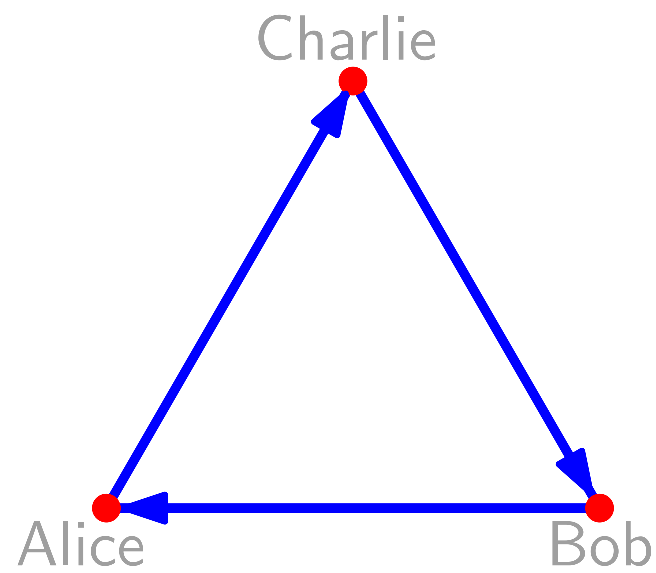


F: Was ist ein Graph?

A$_1$: Ein (ungerichteter) Graph ist ein Tupel (V, E), wobei
- V *Knotenmenge* und
- $E \subseteq \binom{V}{2} = \{\{u, v\} \subseteq V \mid u \neq v\}$ *Kantenmenge*.

$V = \{000, 001, \ldots, 111\}$
$\{u, v\} \in E \quad \Leftrightarrow \quad ?$

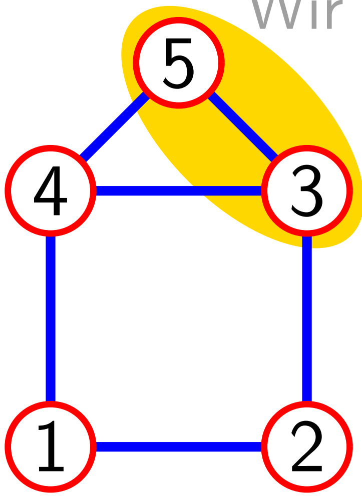




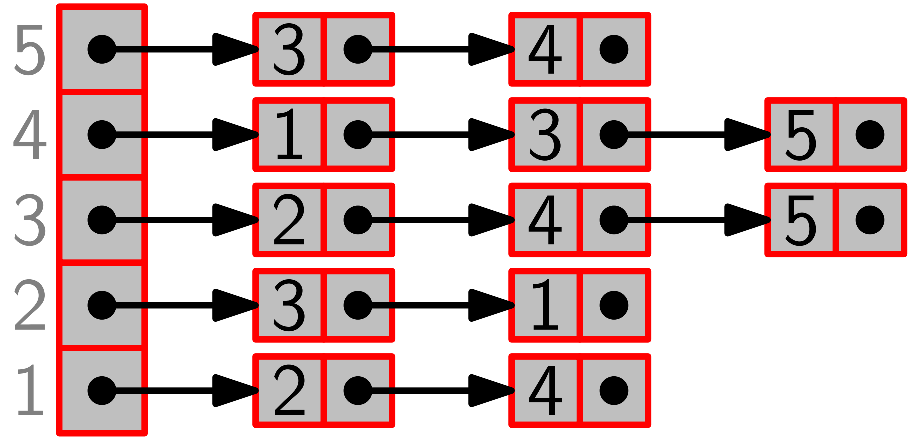
A$_2$: Ein *gerichteter* Graph ist ein Tupel (V, E), wobei
- V *Knotenmenge* und
- $E \subseteq V \times V = \{(u, v) \in V^2 \mid u \neq v\}$ *Kantenmenge*.

F: Wie repräsentiere ich einen Graphen?

Wir sagen: Knoten 3 und 5 sind *adjazent*.



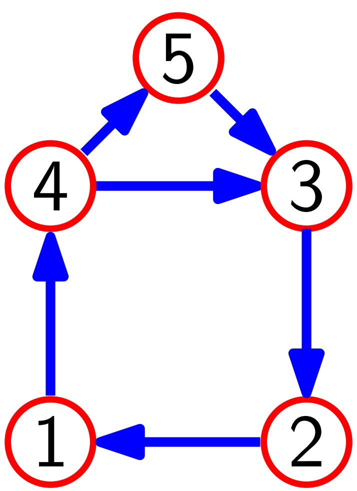
ungerichteter Graph



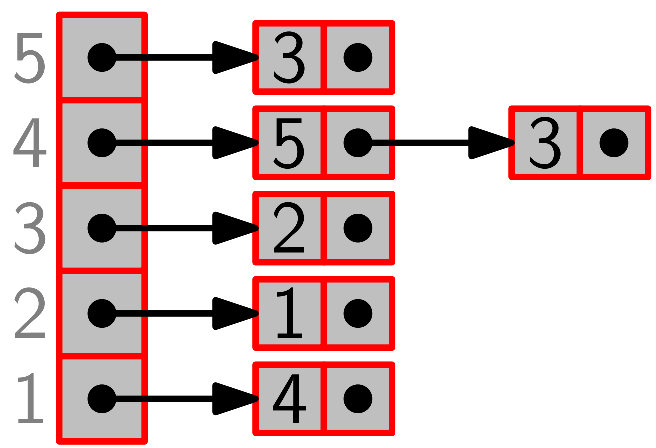
Adjazenzlisten

$$\begin{array}{c|ccccc|} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 & 1 & 1 \\ 4 & 1 & 0 & 1 & 0 & 1 \\ 5 & 0 & 0 & 1 & 1 & 0 \\ \hline \end{array}$$

Adjazenzmatrix



gerichteter Graph



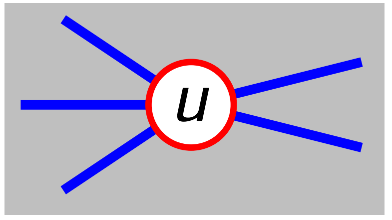
$\mathrm{Adj}[i] = \{j \in V \mid (i,j) \in E\}$

$$\begin{array}{c|ccccc|} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 & 0 & 0 \\ 3 & 0 & 1 & 0 & 0 & 0 \\ 4 & 0 & 0 & 1 & 0 & 1 \\ 5 & 0 & 0 & 1 & 0 & 0 \\ \hline \end{array}$$

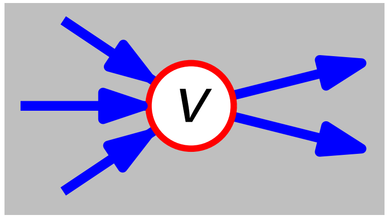
$a_{ij} = 1 \Leftrightarrow (i,j) \in E$

Grad eines Knotens

Def.



$$\deg(u) = |\mathrm{Adj}[u]|$$



$$\mathrm{outdeg}(v) = |\mathrm{Adj}[v]|$$
$$\mathrm{indeg}(v) = |\{u \in V : (u, v) \in E\}|$$

Beob. Sei $G = (V, E)$ ein ungerichteter Graph.
Dann ist die Summe aller Knotengrade $= \; 2 \cdot |E|$.

Beweis. Technik des *zweifachen Abzählens:*

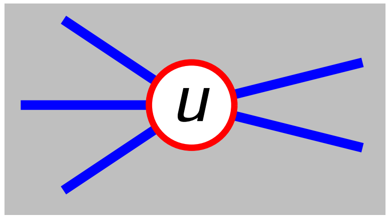
Zähle alle Knoten-Kanten-Inzidenzen.

Eine Kante ist *inzident* zu ihren Endknoten.

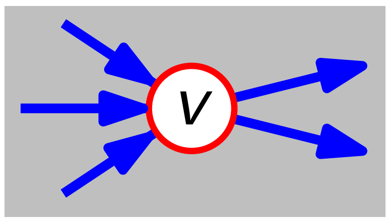
Ein Knoten ist *inzident* zu allen Kanten, deren Endknoten er ist.

Grad eines Knotens

Def.



$\deg(u) = |\mathrm{Adj}[u]|$



$\mathrm{outdeg}(v) = |\mathrm{Adj}[v]|$

$\mathrm{indeg}(v) = |\{u \in V : (u, v) \in E\}|$

Beob. Sei $G = (V, E)$ ein ungerichteter Graph.
Dann ist die Summe aller Knotengrade $= 2 \cdot |E|$.

Beweis. Technik des *zweifachen Abzählens:*

Zähle alle Knoten-Kanten-Inzidenzen.

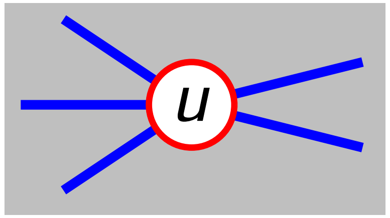
Aus Sicht der Knoten: $\sum_{v \in V} \deg v$

Aus Sicht der Kanten: $\sum_{e \in E} 2 = 2 \cdot |E|$

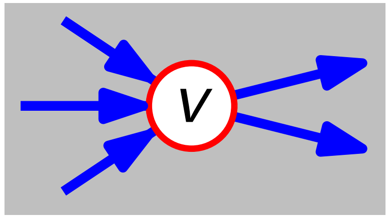
also gleich!

Grad eines Knotens

Def.



$$\deg(u) = |\mathrm{Adj}[u]|$$



$$\mathrm{outdeg}(v) = |\mathrm{Adj}[v]|$$
$$\mathrm{indeg}(v) = |\{u \in V : (u, v) \in E\}|$$

Beob. Sei $G = (V, E)$ ein ungerichteter Graph.
Dann ist die Summe aller Knotengrade $= 2 \cdot |E|$.

Sätzle. Die Anzahl der Knoten ungeraden Grades ist gerade.

Beweis. $\underbrace{\boxed{2 \cdot |E|}}_{\textit{gerade!}} = \underbrace{\boxed{\sum_{v \in V} \deg v}}_{\textit{gerade!}} = \underbrace{\boxed{\sum_{v \in V_{\mathrm{ger}}} \deg v}}_{\textit{gerade!}} + \underbrace{\boxed{\sum_{v \in V_{\mathrm{ung}}} \deg v}}_{\Rightarrow \ \textit{gerade!}}$

$\boxed{\sum_{v \in V_{\mathrm{ung}}} \deg v}$ *gerade* $\Rightarrow$ $|V_{\mathrm{ung}}|$ ist *gerade!* $\square$