# Algorithmische Graphentheorie

Sommersemester 2022

1. Vorlesung

## Graphen: Eine Einführung

*Lehrstuhl für Informatik I* *Alexander Wolff*

# Termine & das Kleingedruckte

## Vorlesung:

– Folien und Videos auf WueCampus
– Mittwochs, 10:15–11:45, Vorlesung im HS 2
  Fragen: mündlich in der VL oder schriftlich im UniWü-Chat.
– Dozent: Prof. Alexander Wolff
  (Büro 01.001, Gebäude M4, Sprechstunde: Mi, 13:30–14:30)

Alle Links dazu auf WueCampus!

## Übungen:

– Organisation: Johannes Zink
  (Büro 01.007, Gebäude M4, dort oder per Email erreichbar)
– TutorInnen:
  Martin Hesse, Antonio Lauerbach, Thanh Mai Pham, Samuel Wolf
– Freitags, 8:30 (Gruppe 1), 10:15 (Gruppen 2+3), 12:15 (Gruppe 4)
– Erstmals schon diese Woche, **29.4.!**

# Termine & das Kleingedruckte

## Übungsaufgaben:

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

– Manche Übungsaufgaben müssen mit OPL bearbeitet werden.
– Dafür gibt es am Fr, 6.5. eine Einführung im CIP-Raum A001 (Gruppen 1, 2, 4) und in A002 (Gruppe 3), wo OPL installiert ist.

# 2x Anmelden!

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

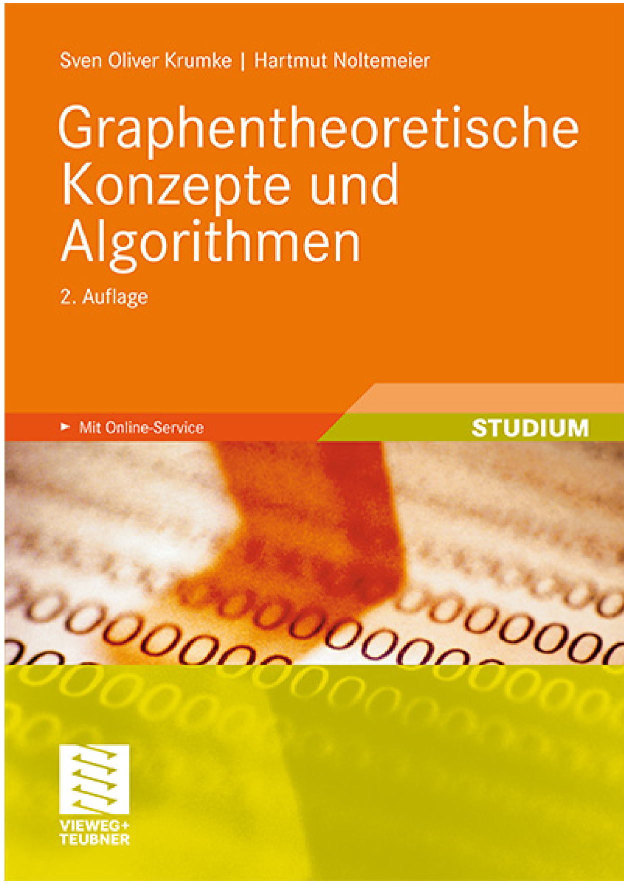
- Übungseinteilung erfolgt über WueStudy
  (Die Übungszeiten sind wichtig für die persönliche Besprechung Ihrer Kleingruppe mit dem Übungsleiter.)
- WueCampus für Kommunikation, z.B. falls eine Übungsaufgabe fehlerhaft oder missverständlich ist.
  (Zum Einschreiben klicken Sie auf das kleine Zahnrad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

**Klausuren:**

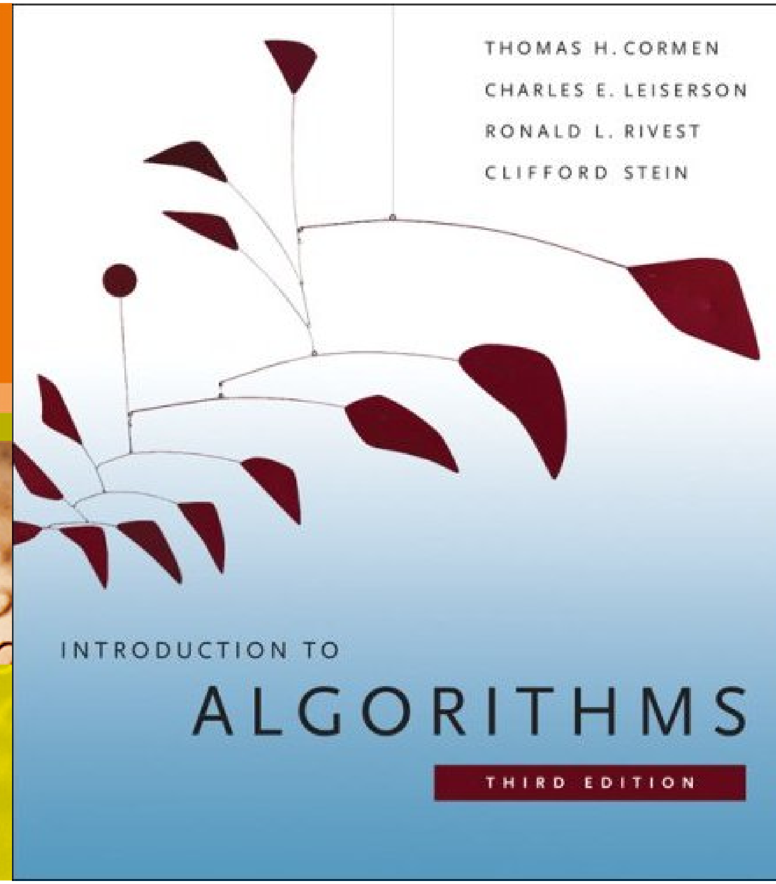
– 1. Termin: 1.8., 12–14 Uhr [Anmeldung 16.04.–15.07.]
– 2. Termin: Anfang Oktober [Anmeldung 01.09.–30.09.]

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

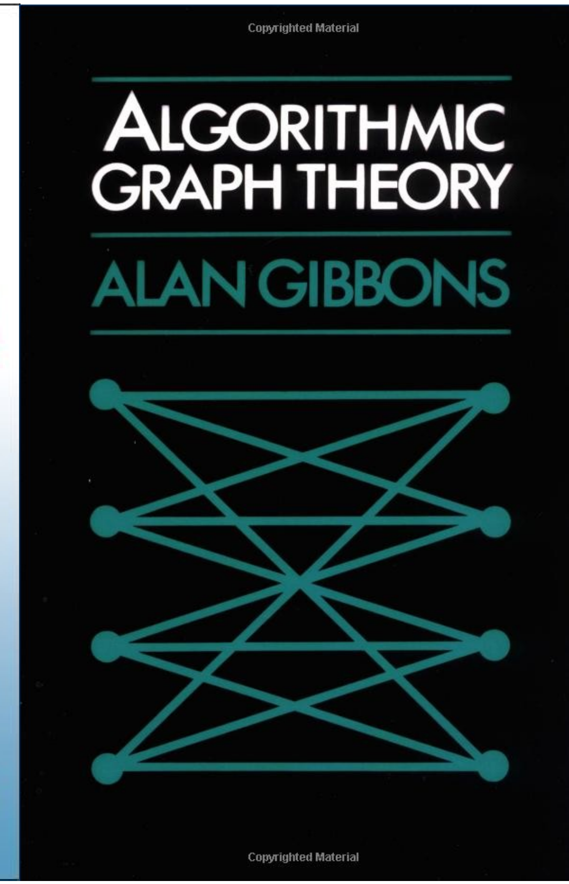
# Bücher zur Vorlesung




[KN]




[CLRS]




[G]

Hauptreferenz; elektronische Kopie über die Unibib erhältlich

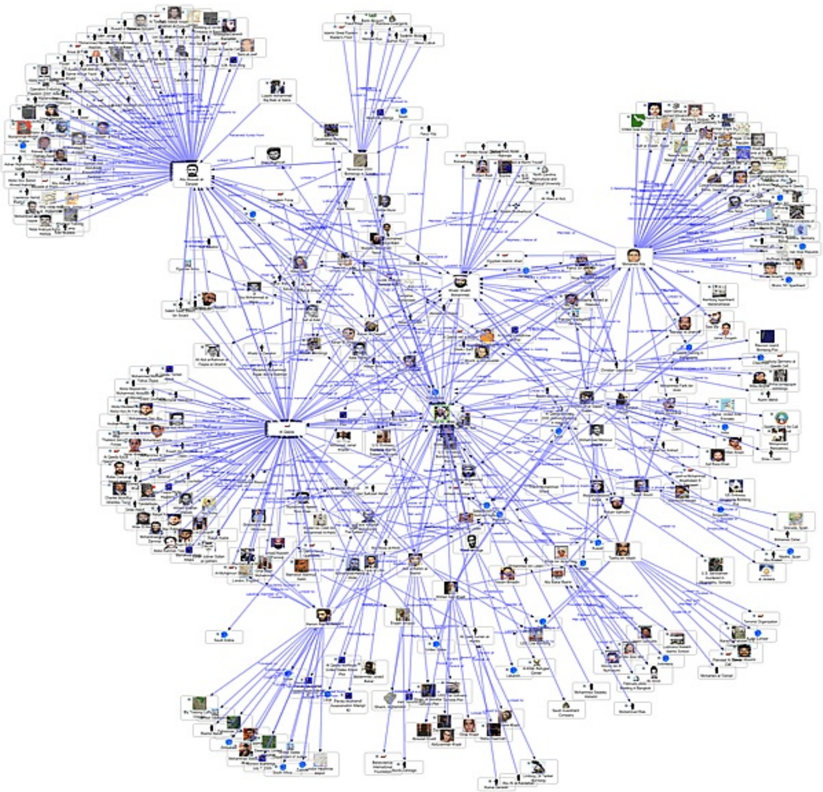
# Voraussetzungen

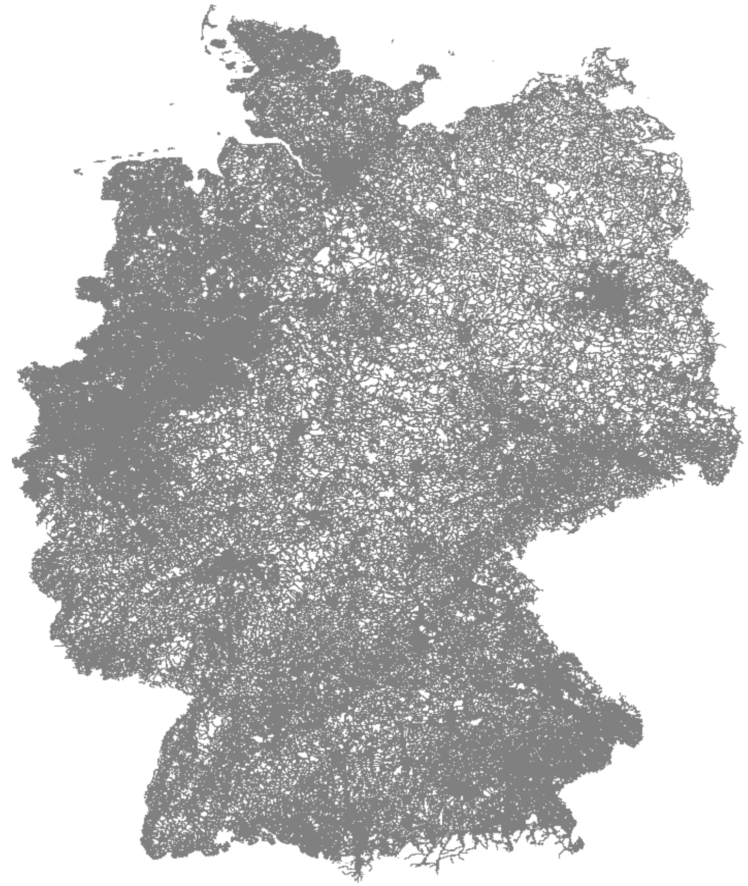
Wissen aus der Vorlesung *Algorithmen und Datenstrukturen*:

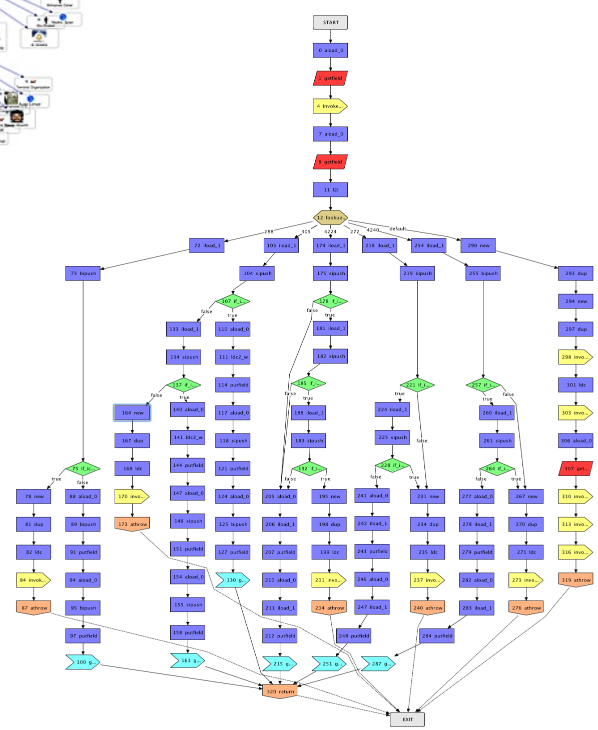
- Graphdurchlauf-Strategien
  - Breitensuche
  - Tiefensuche
- Berechnung kürzester Wege
  - Breitensuche
  - Algorithmus von Dijkstra
- Minimale Spannbäume
  - Algorithmus von Kruskal
  - Algorithmus von Jarník–Prim

Repetitorium in der allerersten Übung

# Graphen







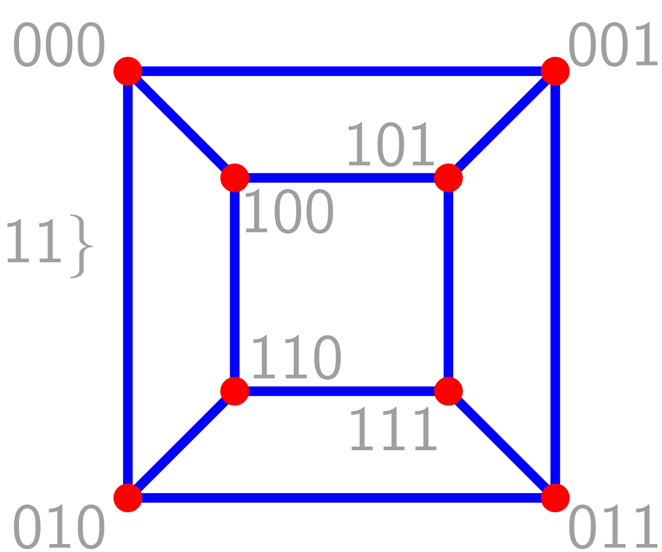
# F: Was ist ein Graph?

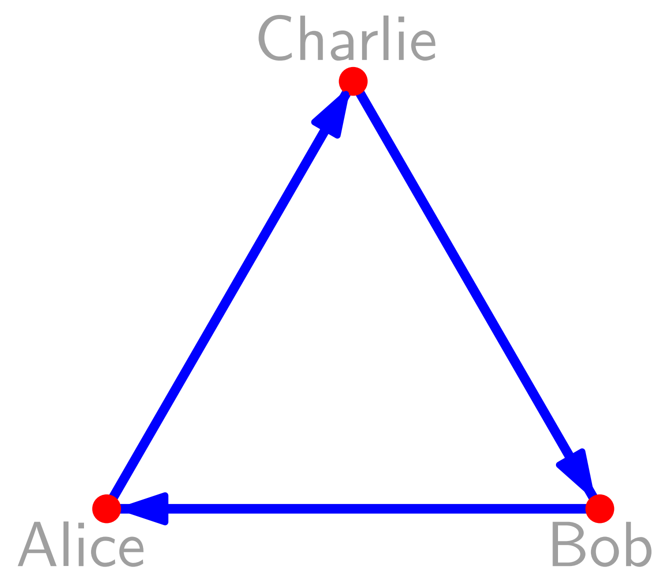
$A_1$: Ein (ungerichteter) Graph $G$ ist ein Tupel $({\color{red}V(G)}, {\color{blue}E(G)})$:

– ${\color{red}V(G)}$ *Knotenmenge* und

– ${\color{blue}E(G)} \subseteq \binom{V}{2} = \{\{u, v\} \subseteq V(G) \mid u \neq v\}$ *Kantenmenge*.

$V(G) = \{000, 001, \ldots, 111\}$
$\{u, v\} \in E(G) \quad \Leftrightarrow \quad {\color{red}?}$



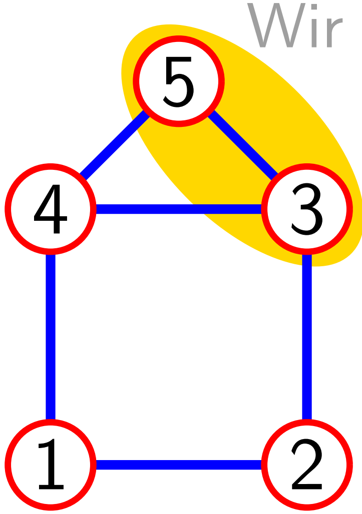

$A_2$: Ein *gerichteter* Graph $G$ ist ein Tupel $({\color{red}V(G)}, {\color{blue}E(G)})$, wobei

– ${\color{red}V(G)}$ *Knotenmenge* und

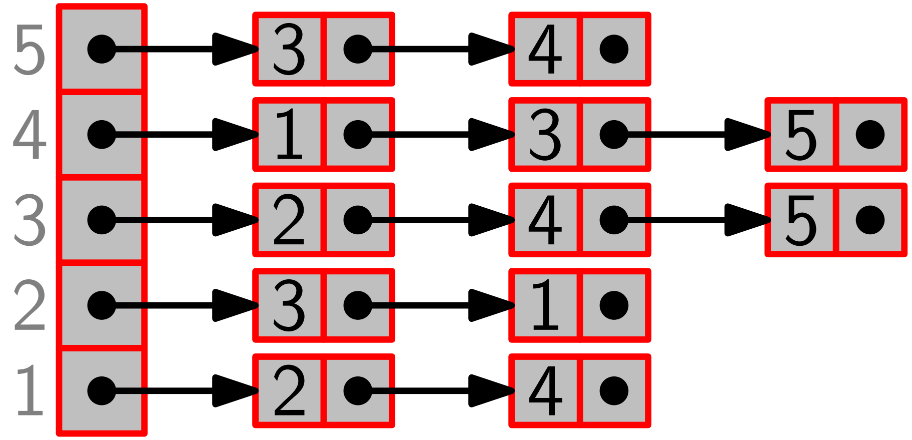
– ${\color{blue}E(G)} \subseteq V(G) \times V(G) = \{(u, v) \mid u, v \in V(G), u \neq v\}$ *Kantenmenge*.

# F: Wie repräsentiere ich einen Graphen?

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



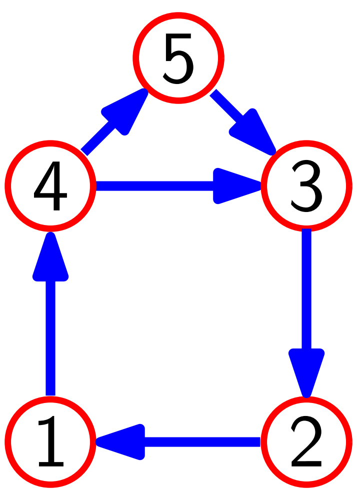

ungerichteter Graph



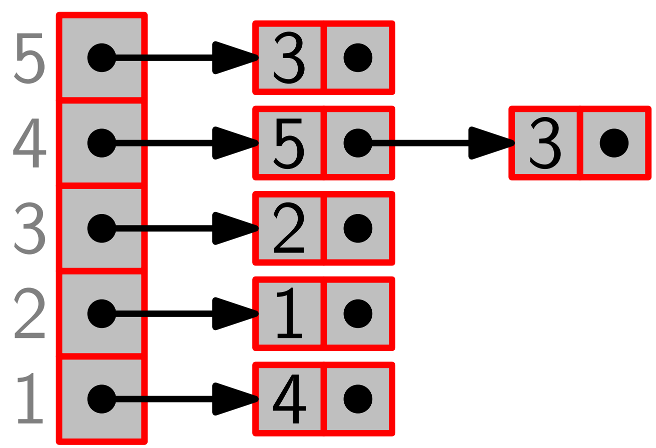

*Adjazenzlisten*

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 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 |

*Adjazenzmatrix*




gerichteter Graph



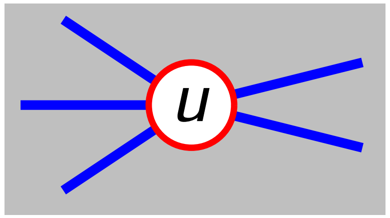

$\mathrm{Adj}[i] = \{j \in V(G) : (i,j) \in E(G)\}$

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 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 |

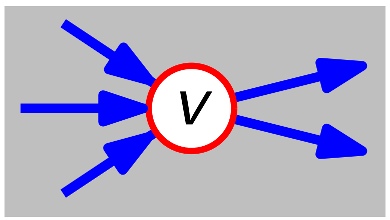
$a_{ij} = 1 \Leftrightarrow (i,j) \in E(G)$

# Grad eines Knotens

**Def.**




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




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

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

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

*Beweis.* Technik des *zweifachen Abzählens:*

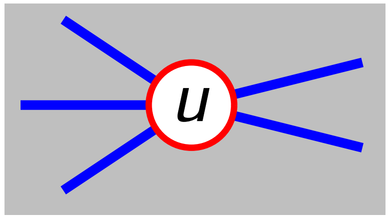
Zähle alle Knoten-Kanten-Inzidenzen.

Eine Kante ist *inzident* zu ihren Endknoten.

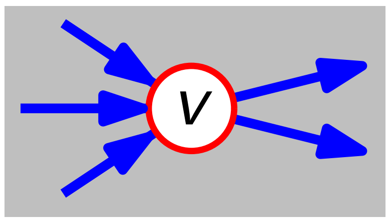
Ein Knoten ist *inzident* zu allen Kanten, deren Endknoten er ist.

# Grad eines Knotens

**Def.**




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




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

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

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

*Beweis.* Technik des *zweifachen Abzählens:*

Zähle alle Knoten-Kanten-Inzidenzen.

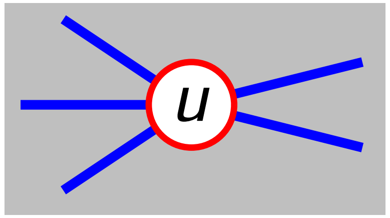
Aus Sicht der Knoten: $\sum_{v \in V(G)} \deg(v)$

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

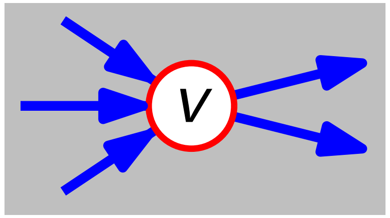
also gleich!

# Grad eines Knotens

**Def.**




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




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

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

**Sätzle.** Die Anzahl der Knoten ungeraden Grades ist gerade.

$V(G) = V_{\mathrm{ger}}(G) \cup V_{\mathrm{ung}}(G)$

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

$\sum_{v \in V_{\mathrm{ung}}(G)} \deg(v)$ *gerade* $\Rightarrow$ $|V_{\mathrm{ung}}(G)|$ *gerade!* $\square$