

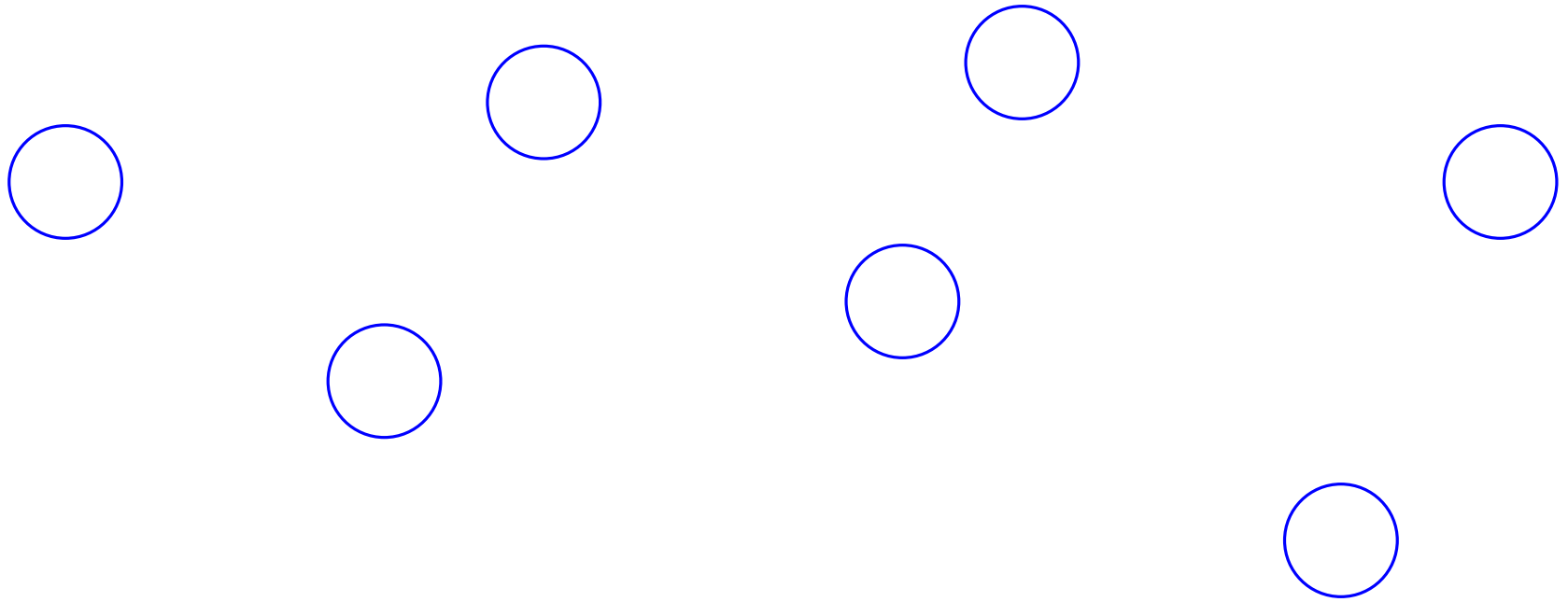
# **Graphen und diskrete Optimierung**

im Bachelorstudiengang 'Informatik und Nachhaltigkeit'

Marie Schmidt

17.04.2023

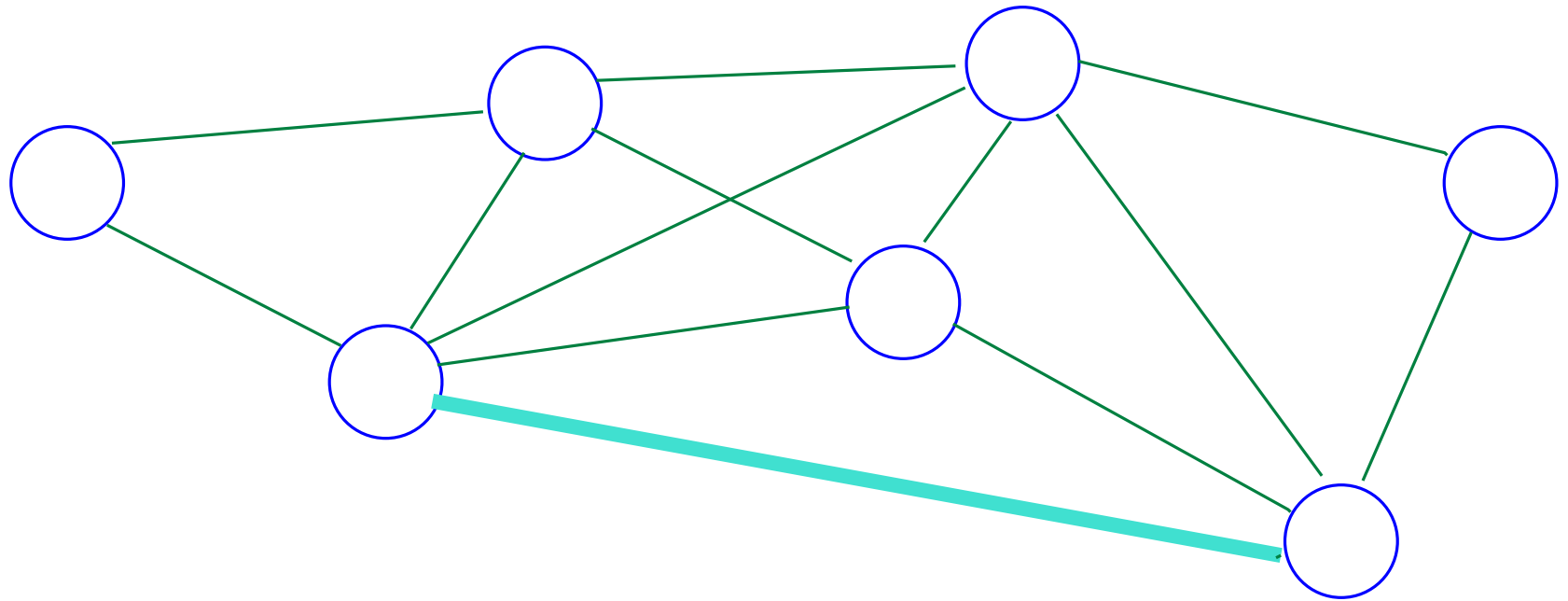
# Graphen



Ein Graph besteht aus

Knoten  $V$

# Graphen ... und Optimierung auf Graphen

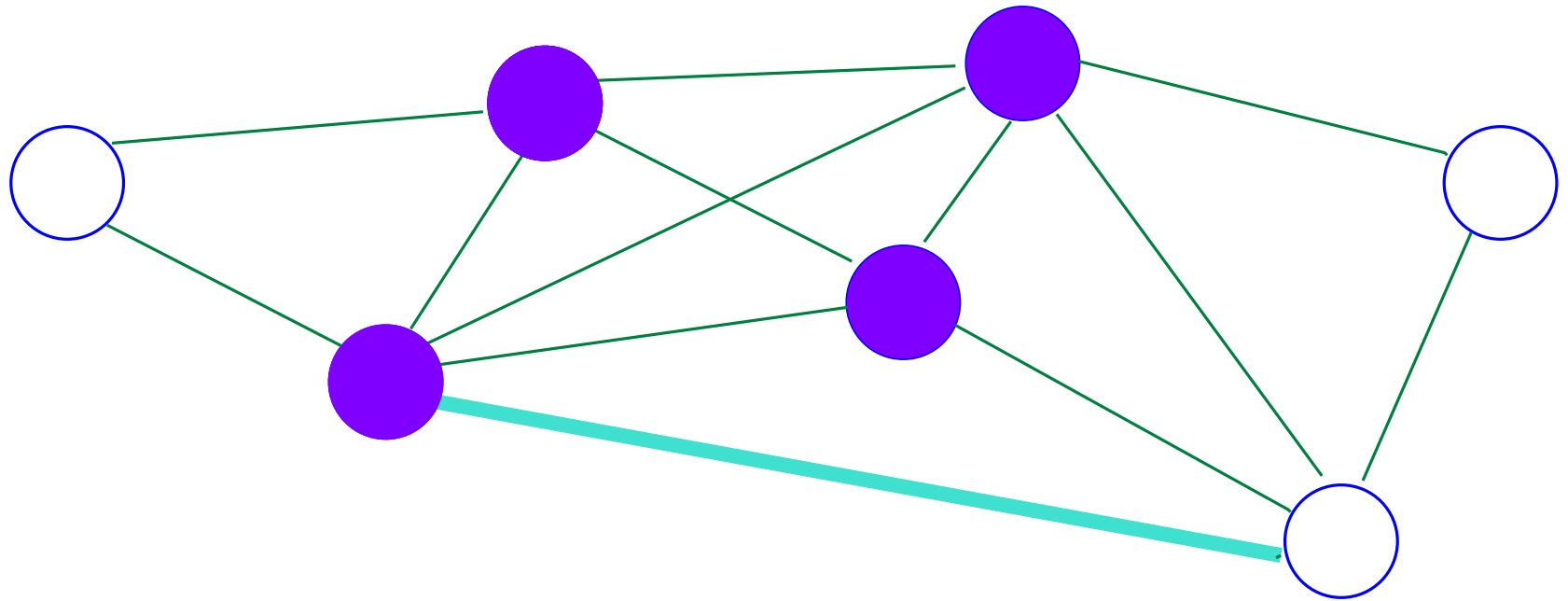


Ein Graph besteht aus

Knoten  $V$

Kanten  $E$

# Graphen ... und Optimierung auf Graphen



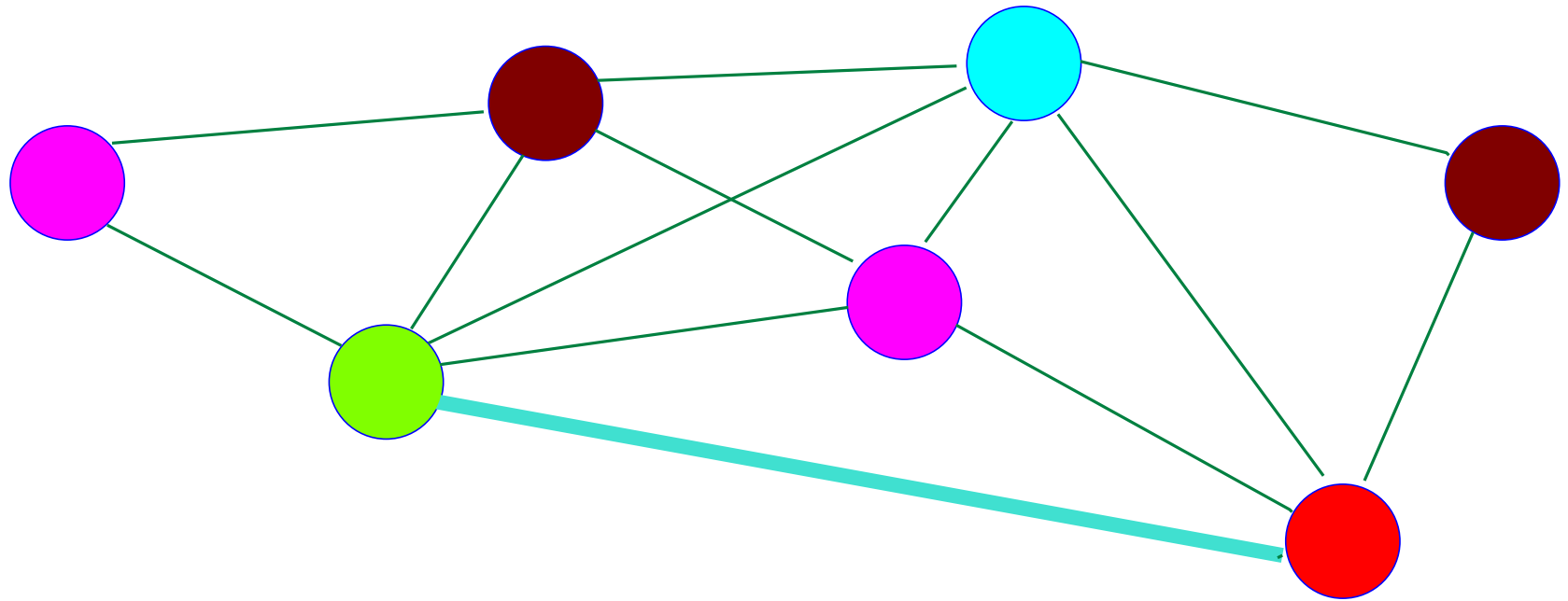
Ein Graph besteht aus

Knoten  $V$

Kanten  $E$

Wie groß ist die größte  
Clique im Graphen?

# Graphen ... und Optimierung auf Graphen



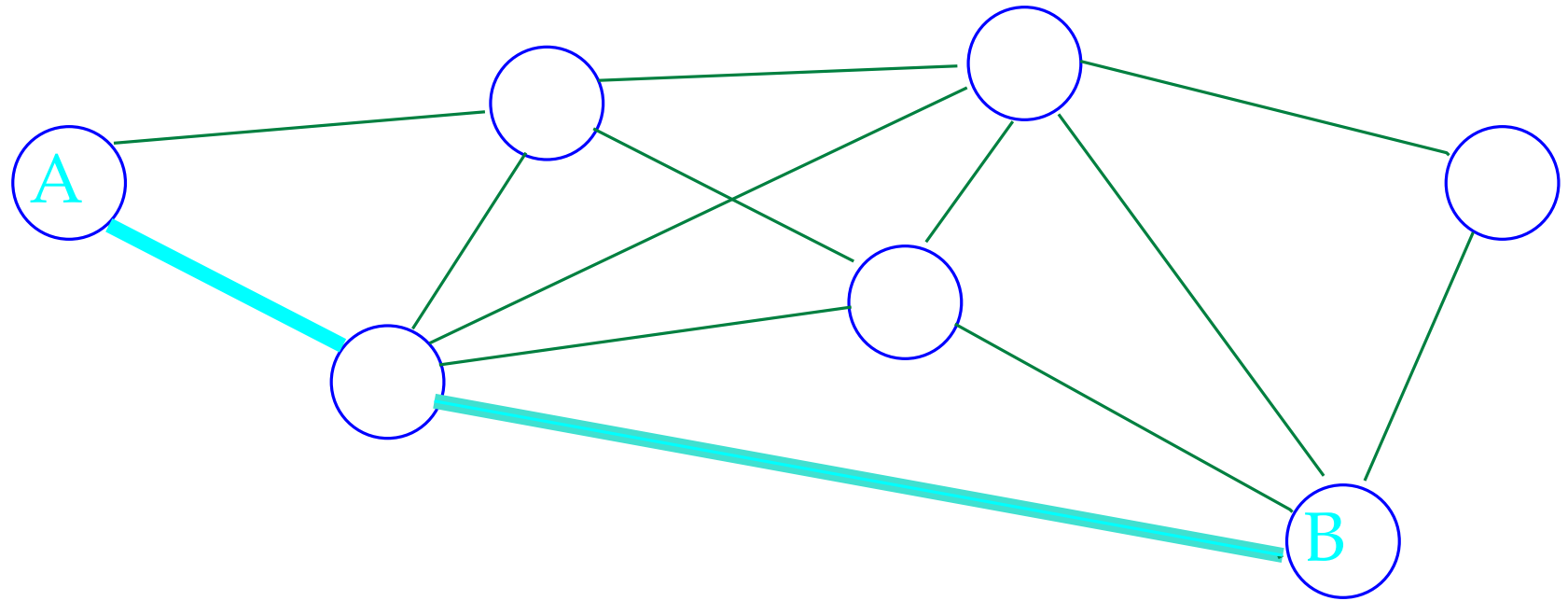
Ein Graph besteht aus

Knoten  $V$

Kanten  $E$

Wie viele Farben braucht man mindestens, um den Graphen so zu färben, dass benachbarte Knoten verschiedene Farben haben?

# Graphen ... und Optimierung auf Graphen



Ein Graph besteht aus

Knoten  $V$

Kanten  $E$

Was ist der kürzeste Weg  
von A nach B?

# Über mich

Marie Schmidt, [marie.schmidt@uni-wuerzburg.de](mailto:marie.schmidt@uni-wuerzburg.de)

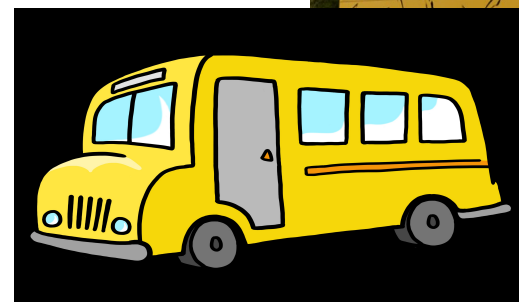
Professorin für Optimierung unter Ressourcenbeschränkungen

# Über mich

Marie Schmidt, [marie.schmidt@uni-wuerzburg.de](mailto:marie.schmidt@uni-wuerzburg.de)

Professorin für Optimierung unter Ressourcenbeschränkungen

Forschung im Bereich Optimierung im öffentlichen Verkehr,  
Transportoptimierung, robuste multikriterielle Optimierung



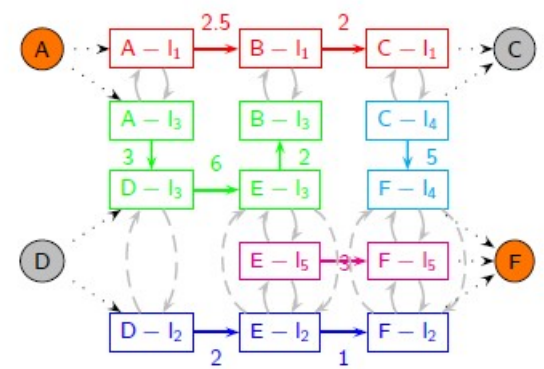
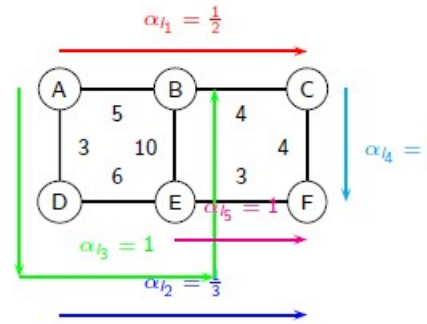
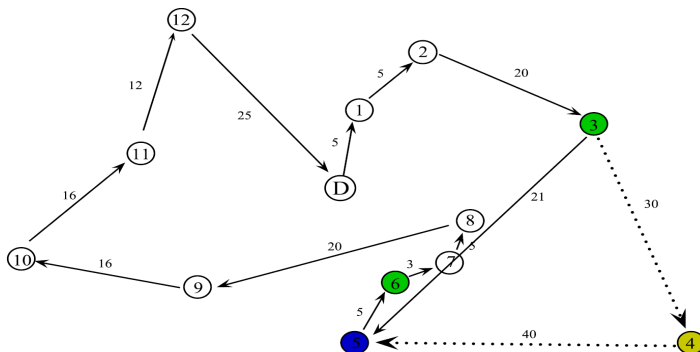
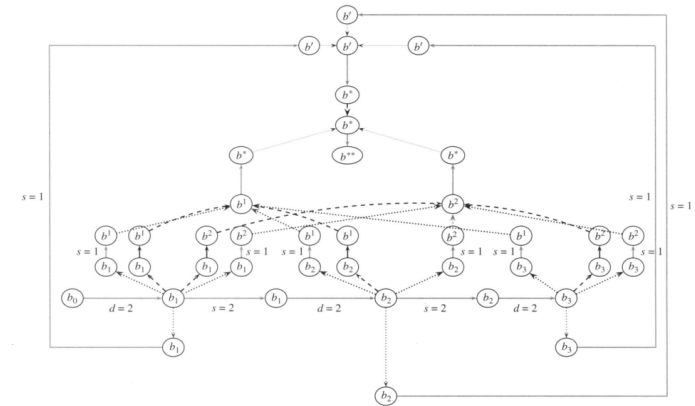
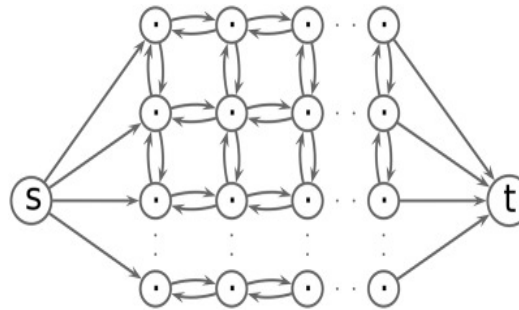
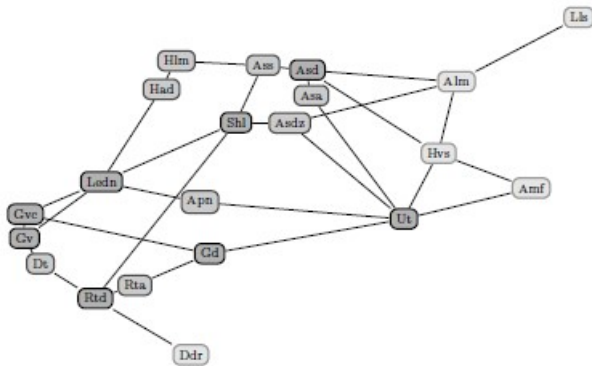


# Über mich

Marie Schmidt, marie.schmidt@uni-wuerzburg.de

Professorin für Optimierung unter Ressourcenbeschränkungen

Forschung im Bereich Optimierung im öffentlichen Verkehr,  
Transportoptimierung, robuste multikriterielle Optimierung



Kurzvorstellung Teilnehmer

und: was sind Ihre Erwartungen an die Vorlesung?

Kurzvorstellung Teilnehmer

und: was sind Ihre Erwartungen an die Vorlesung?

Worum soll er hier gehen?

|

Worum soll er hier gehen?

Strukturen auf Graphen

|

# Worum soll er hier gehen?

Strukturen auf Graphen

Modellieren  
mit Graphen

|

# Worum soll er hier gehen?

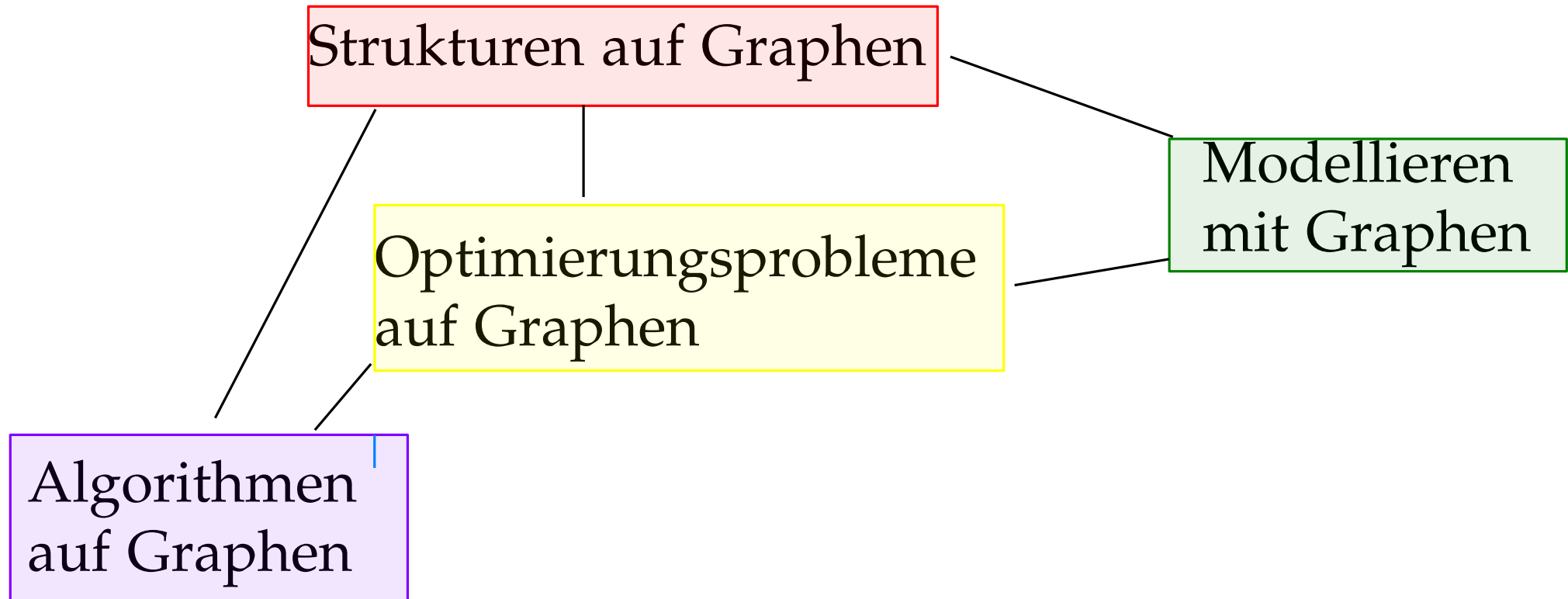
Strukturen auf Graphen

Optimierungsprobleme  
auf Graphen

Modellieren  
mit Graphen

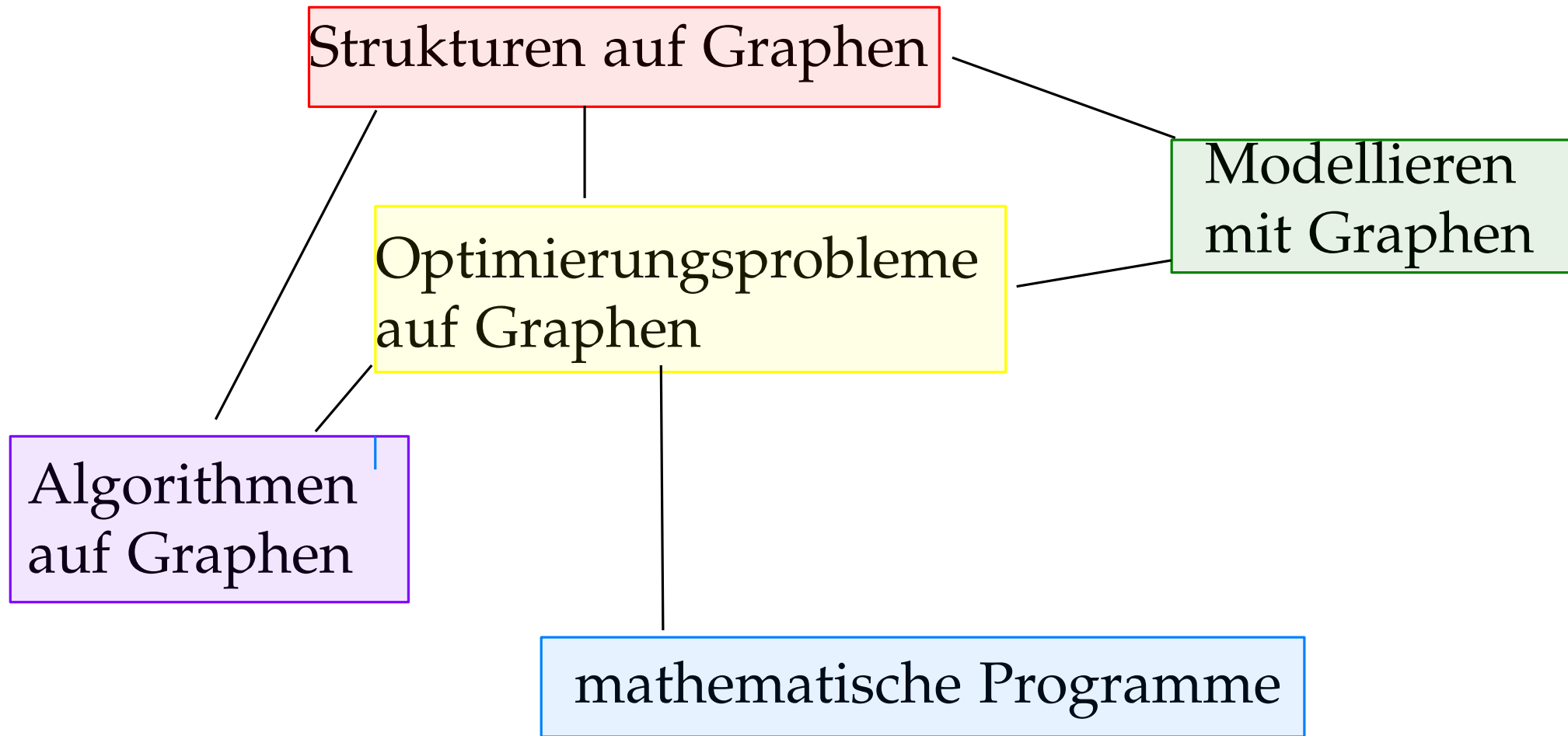
|

# Worum soll er hier gehen?

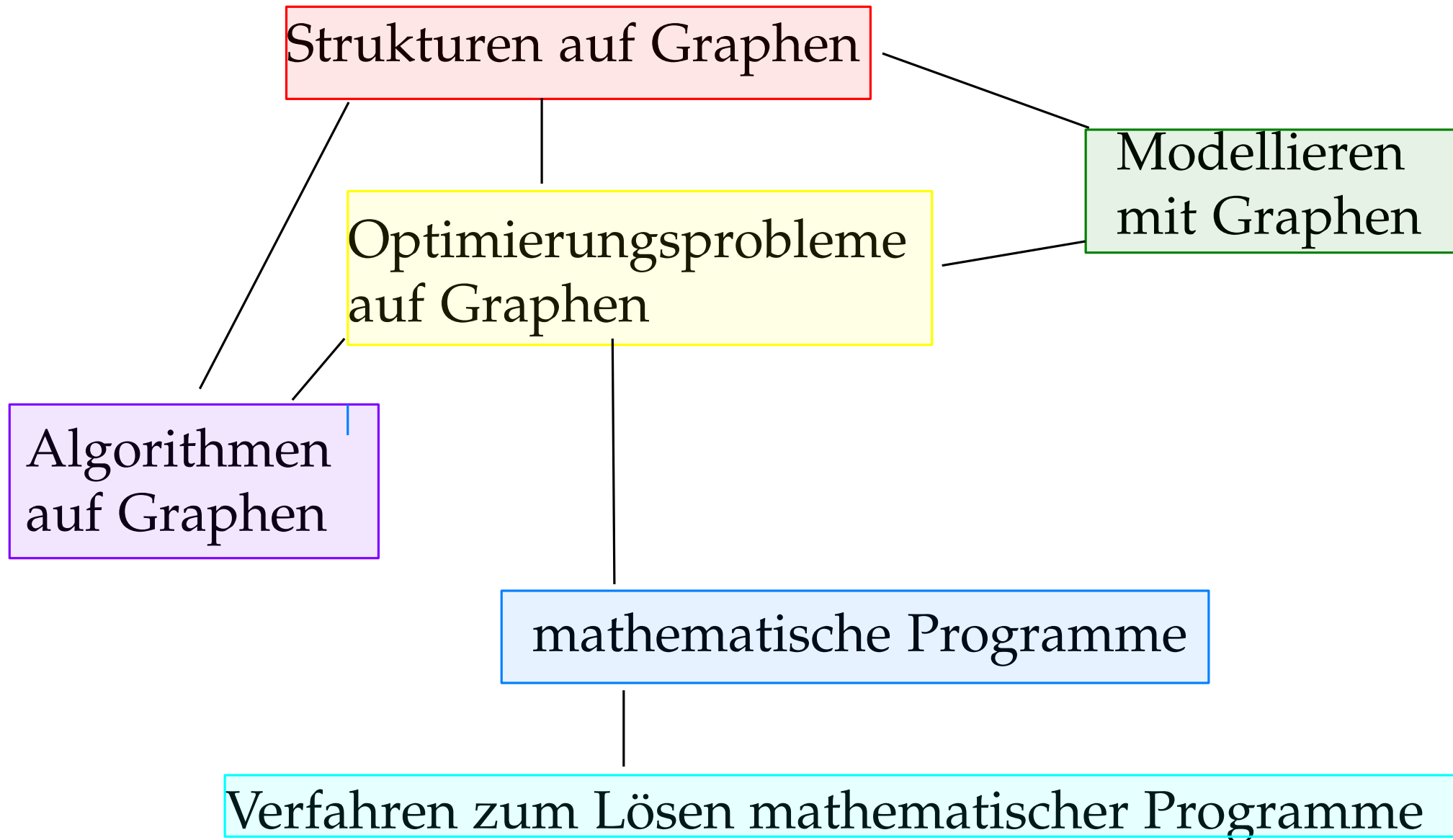




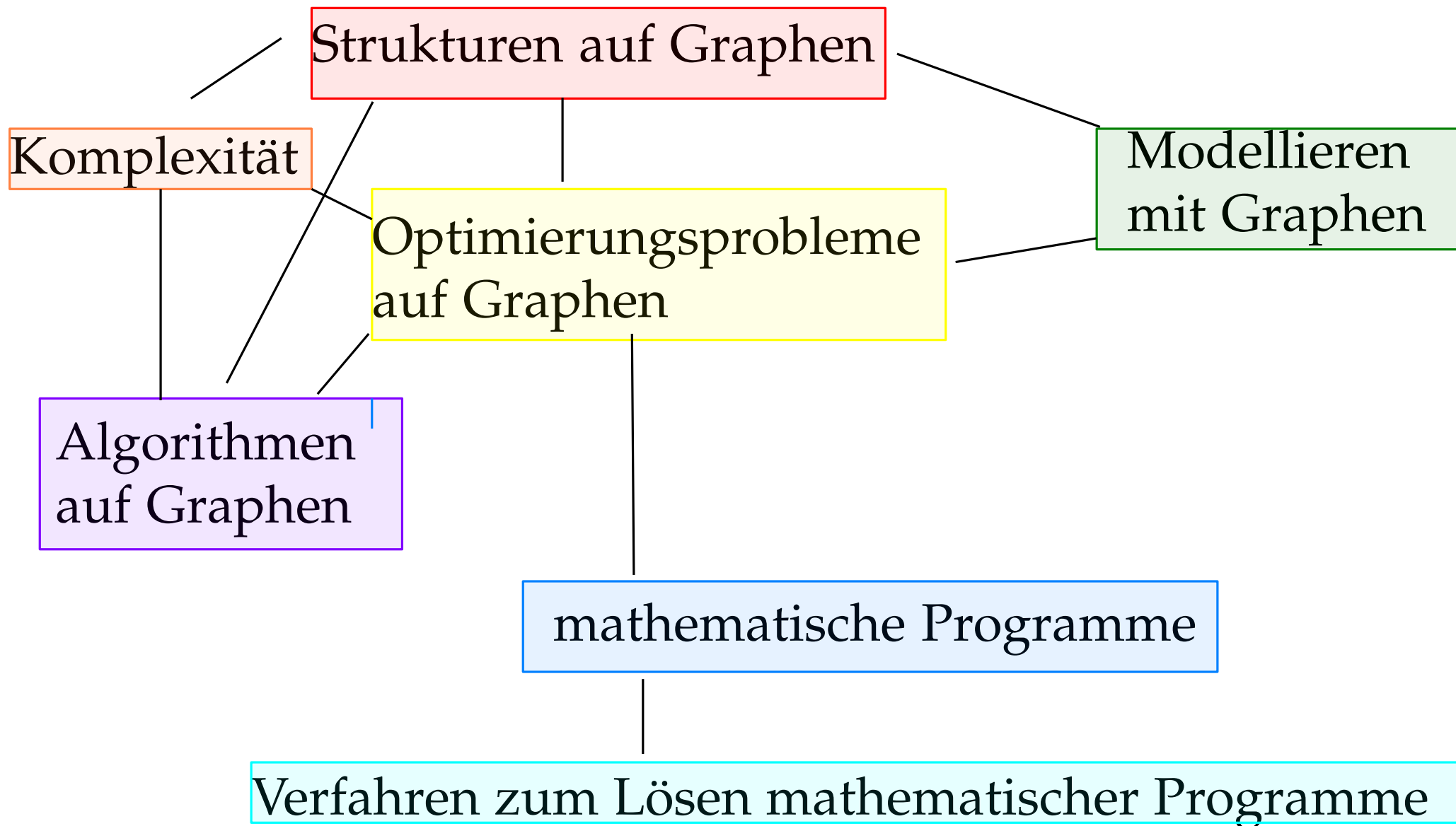
# Worum soll er hier gehen?



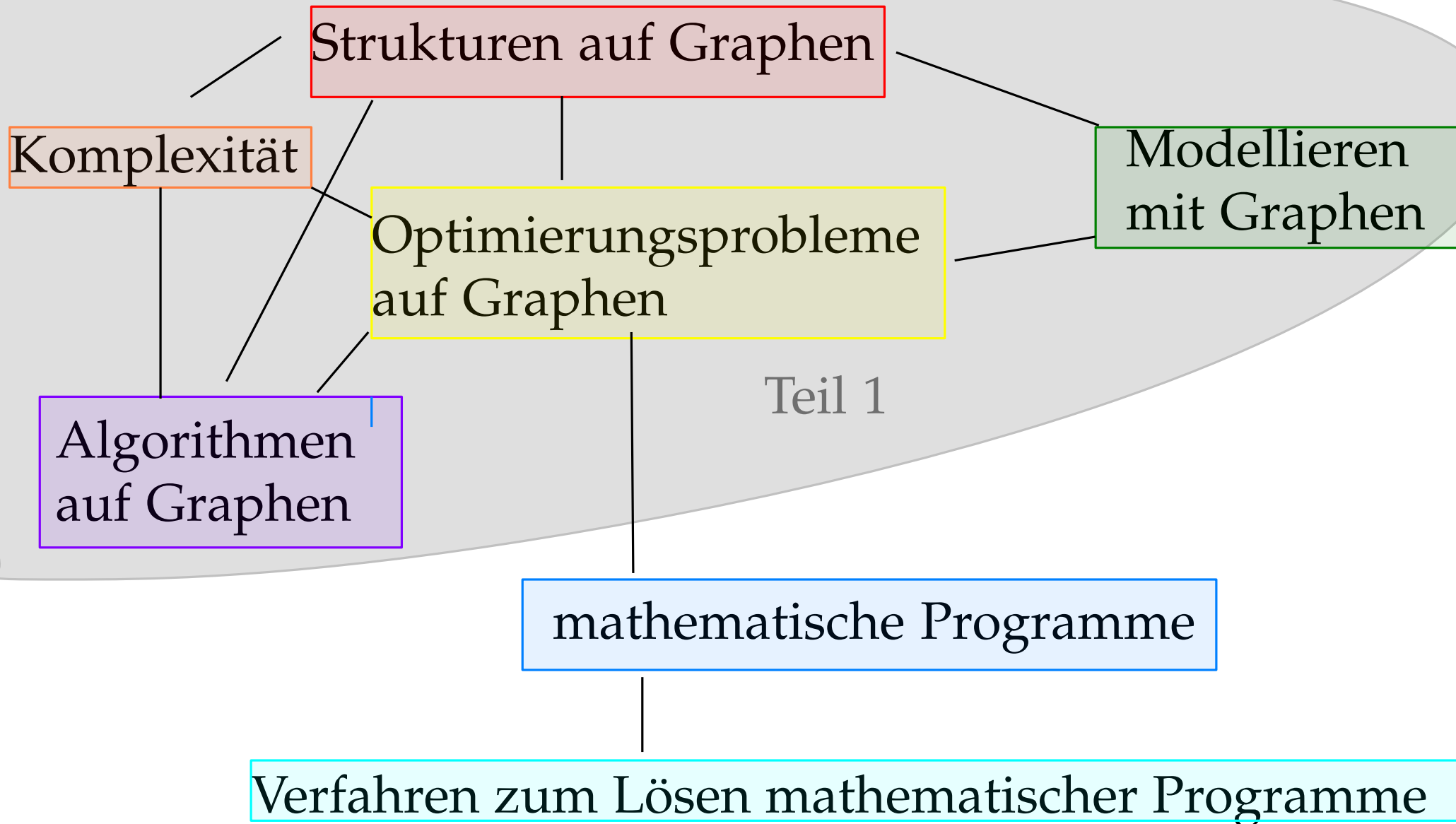
# Worum soll er hier gehen?



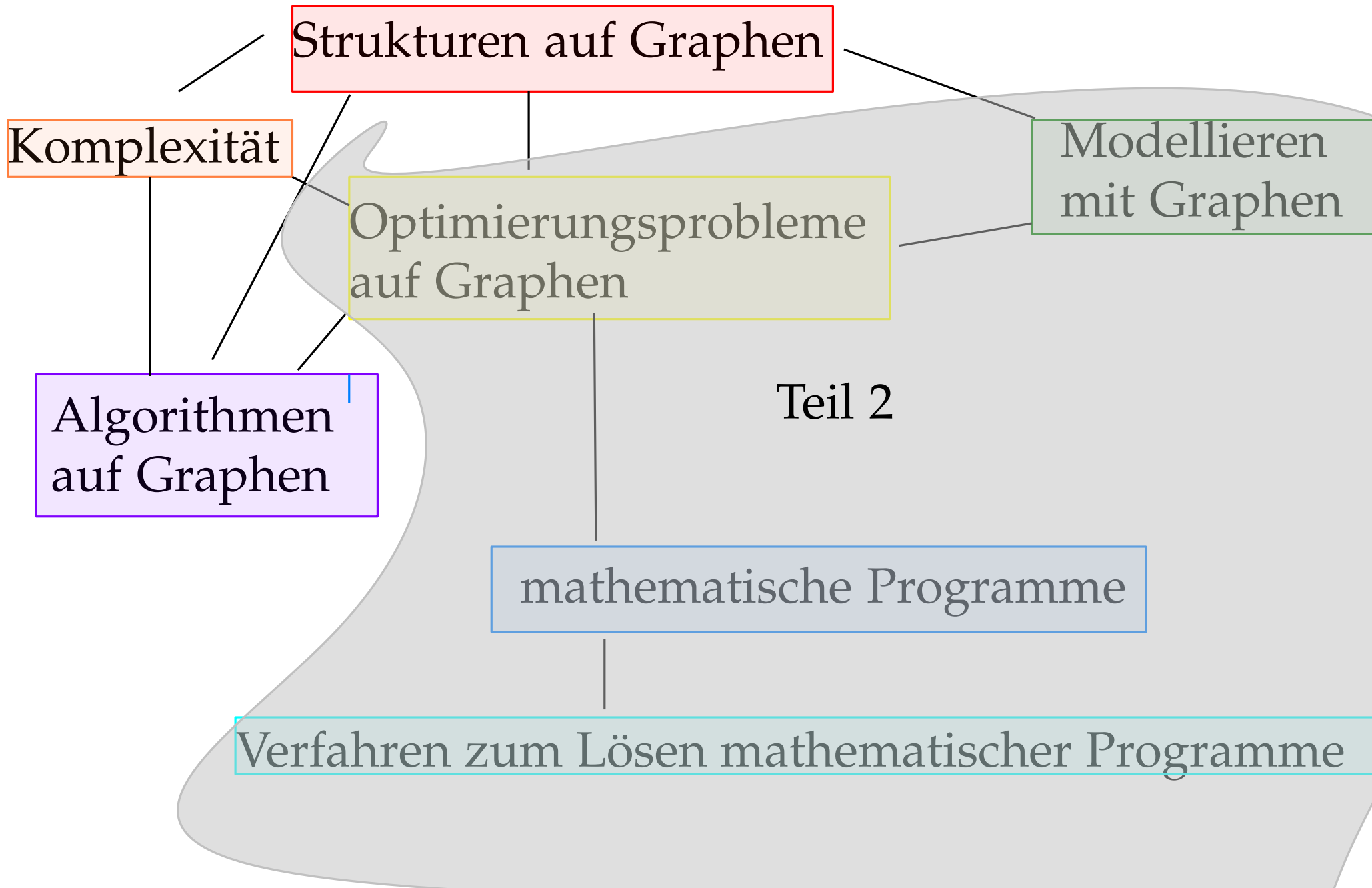
# Worum soll er hier gehen?



# Worum soll er hier gehen?



# Worum soll er hier gehen?



# Agenda

1. Worum es hier geht ✓
2. **Formales und Organisatorisches**
3. Graphen - erste Definitionen und Konzepte
4. Modellieren mit Graphen

# Termine

## **Vorlesung:**

- mittwochs 10:15 im SE10, soweit nicht anders angekündigt
- Pause?
- Folien im Anschluss auf WueCampus
- Dozentin: Marie Schmidt
- Büro 01.018, Gebäude M4, Sprechstunde: Mi, 14:00-15:00  
→ Achtung: heute nur bis 14:30!
- inhaltliche Fragen: in der Vorlesung, Übung, oder im WUECampus  
Diskussionsforum

# Termine

## **Vorlesung:**

- mittwochs 10:15 im SE10, soweit nicht anders angekündigt
- Pause?
- Folien im Anschluss auf WueCampus
- Dozentin: Marie Schmidt
- Büro 01.018, Gebäude M4, Sprechstunde: Mi, 14:00-15:00  
→ Achtung: heute nur bis 14:30!
- inhaltliche Fragen: in der Vorlesung, Übung, oder im WUECampus  
Diskussionsforum

## **Übungen:**

- freitags 8:15-9:45 im Ü1 (Gruppe 1) und 14:15-15:45 im SE1
- Übungsleitung und Organisation: Kendra Reiter,  
kendra.reiter@uni-wuerzburg.de, Büro: Gebäude M4



## Übungsaufgaben:

- Bearbeitung in Gruppen von max. je zwei TeilnehmerInnen
- Ausgabe: mittwochs via WueCampus (erstes Übungsblatt heute!)
- Abgabe: bis mittwochs 10:00 Uhr, auf WueCampus (nur pdf)

Bitte möglichst mit Latex schreiben (Vorlage auf WueCampus)!

Schöne Latex-Bilder (von Graphen) kann man mit ipe machen

(<https://ipe.otfried.org/>), handgezeichnet und ins pdf eingefügt

tut's aber auch

2x Anmelden!

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

- Übungseinteilung

- Anmelden in WueCampus ist wichtig für Kommunikation:  
Mitteilungen und Diskussionsforum

(Zum Einschreiben klicken Sie auf das kleine Zahnrad oben links und wählen dann: 'Mich in diesen Kurs einschreiben'.)

# Prüfung

Entscheidung: Klausur oder mündliche Prüfung und  
Termine bis nächste Woche

# Prüfung

Entscheidung: Klausur oder mündliche Prüfung und Termine bis nächste Woche

Übungszettel bonusfähig:

$\geq 50\%$  der Punkte auf den Übungszetteln?

$\Rightarrow +0.3$  auf Prüfungsnote - falls bestanden

# Prüfung

Entscheidung: Klausur oder mündliche Prüfung und Termine bis nächste Woche

Übungszettel bonusfähig:

$\geq 50\%$  der Punkte auf den Übungszetteln?

$\Rightarrow +0.3$  auf Prüfungsnote - falls bestanden

Achtung: Wenn Sie sich nicht fristgerecht bei WueStudy anmelden, ist es für uns unmöglich, Ihre Note zu verbuchen

# Literaturempfehlung

für den ersten Teil der Vorlesung:

Krumke und Noltemeier, 'Graphentheoretische Konzepte  
und Algorithmen'

verfügbar als pdf über die Seite der Unibibliothek

# Voraussetzungen

Wissen aus der Vorlesung *Algorithmen und Datenstrukturen*:

- Graphdurchlauf-Strategien
- Berechnung kürzester Wege
- Minimale Spannbäume

# 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 Jarnik-Prim



# Voraussetzungen

Wissen aus der Vorlesung *Algorithmen und Datenstrukturen*:

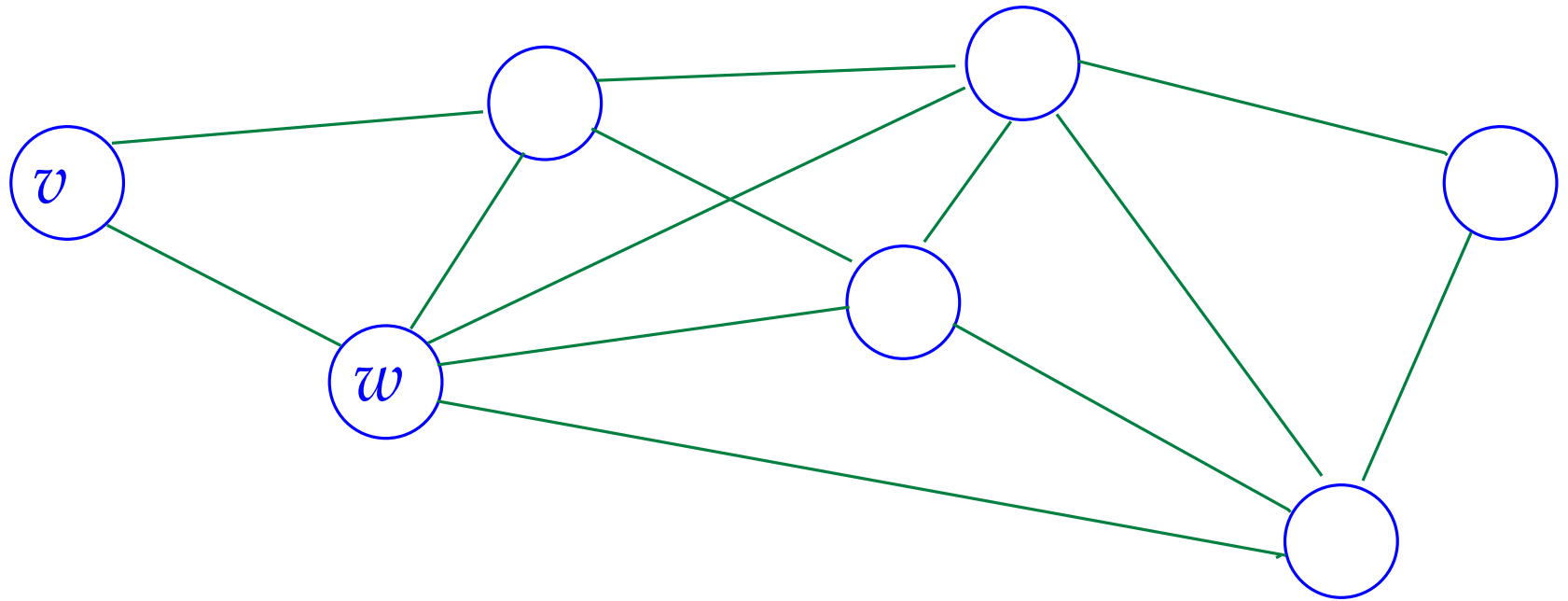
- Graphdurchlauf-Strategien
  - Breitensuche
  - Tiefensuche
- Berechnung kürzester Wege
  - Breitensuche
  - Algorithmus von Dijkstra
- Minimale Spannbäume
  - Algorithmus von Kruskal
  - Algorithmus von Jarnik-Prim

Wiederholung hiervon in der ersten Übung

# Agenda

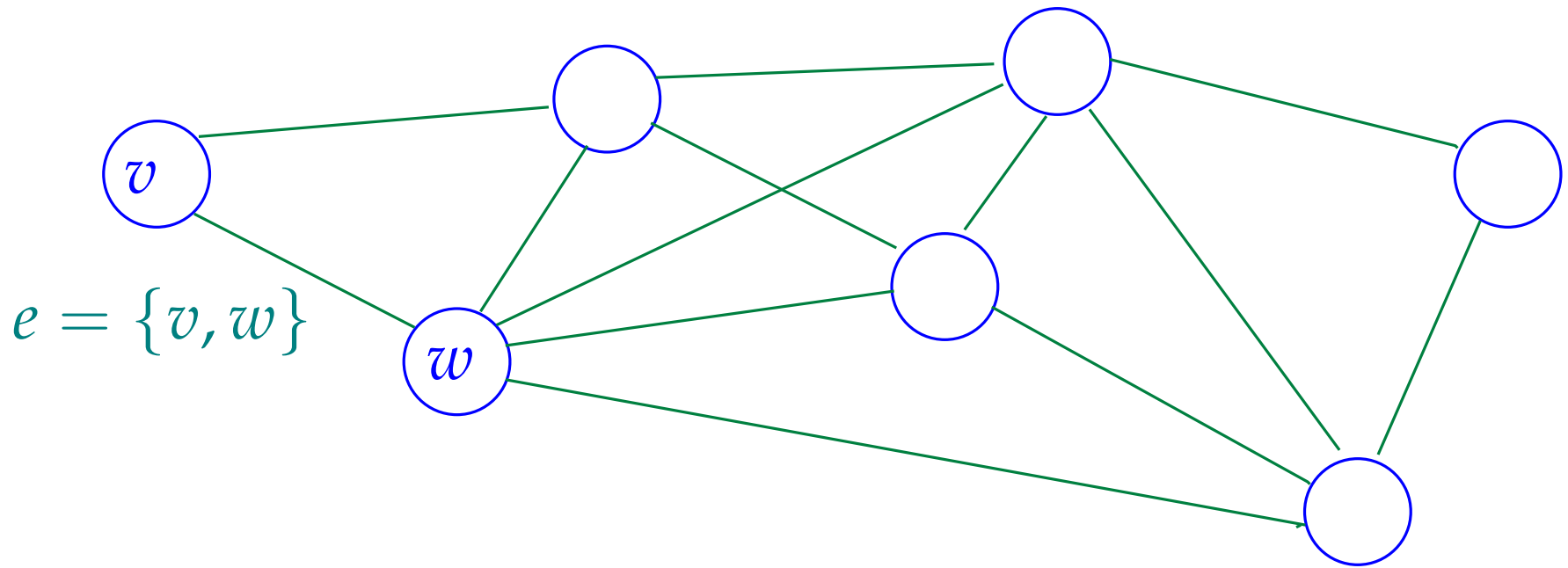
1. Worum es hier geht ✓
2. Formales und Organisatorisches ✓
3. **Graphen - erste Definitionen und Konzepte**
4. Modellieren mit Graphen

# Graphen



Ein (ungerichteter) **Graph**  $G = (V, E)$  besteht aus  
**Knoten**  $V$  (Englisch: 'nodes' oder 'vertices')

# Graphen

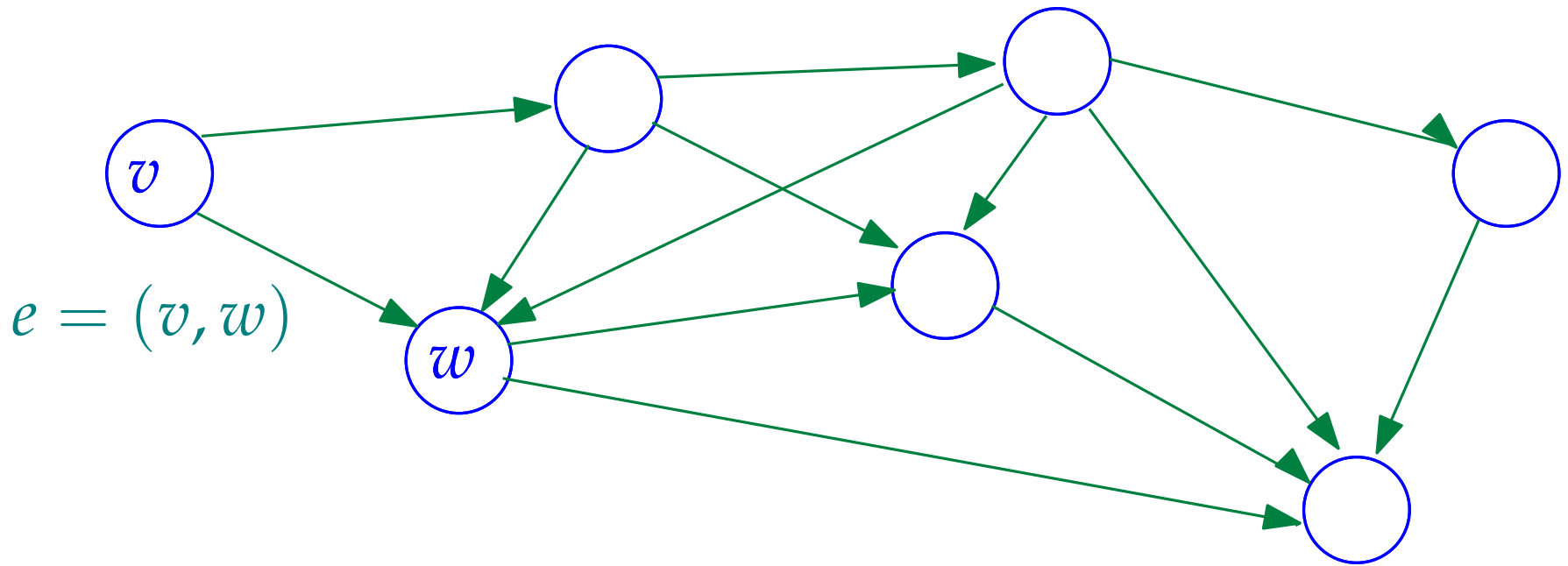


Ein (ungerichteter) **Graph**  $G = (V, E)$  besteht aus

**Knoten**  $V$  (Englisch: 'nodes' oder 'vertices')

Knotenpaaren  $E$ , die als **Kanten** bezeichnet werden  
(Englisch: 'edge' oder 'arc')

# Graphen



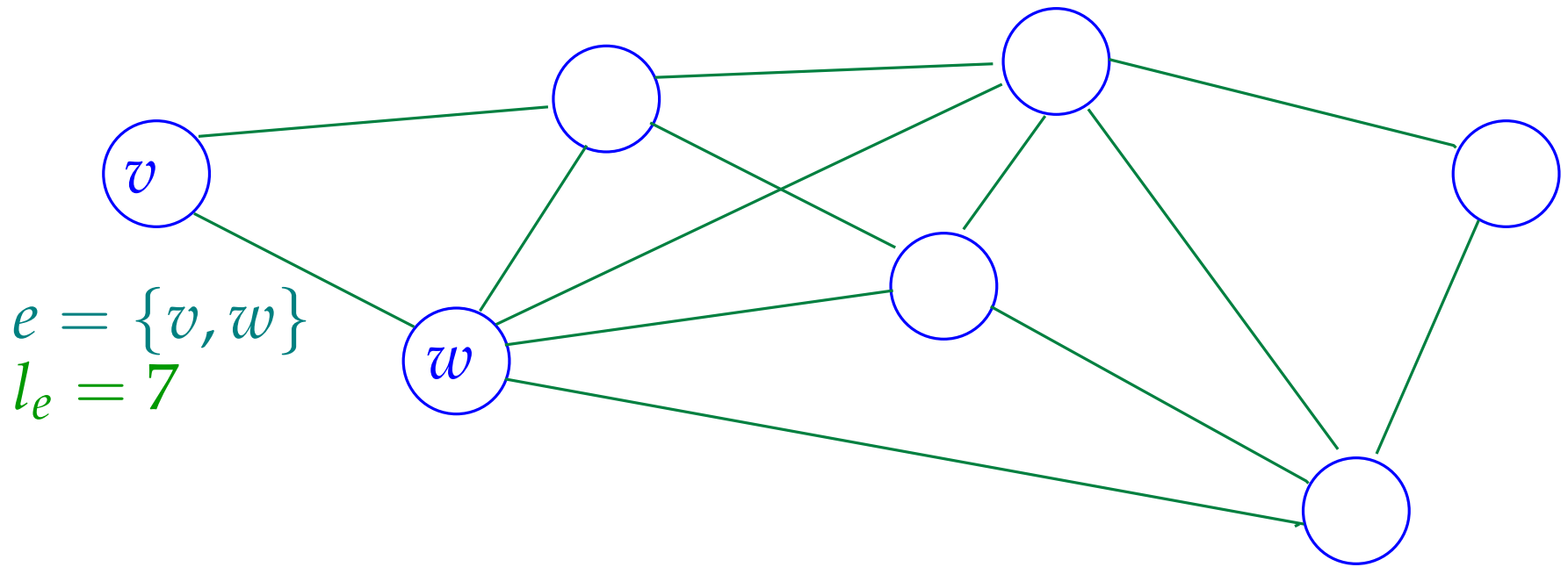
Ein **gerichteter Graph**  $G = (V, E)$  besteht aus

**Knoten**  $V$  (Englisch: 'nodes' oder 'vertices')

Gerichteten Knotenpaaren  $E$ , die als **gerichtete Kanten** bezeichnet werden

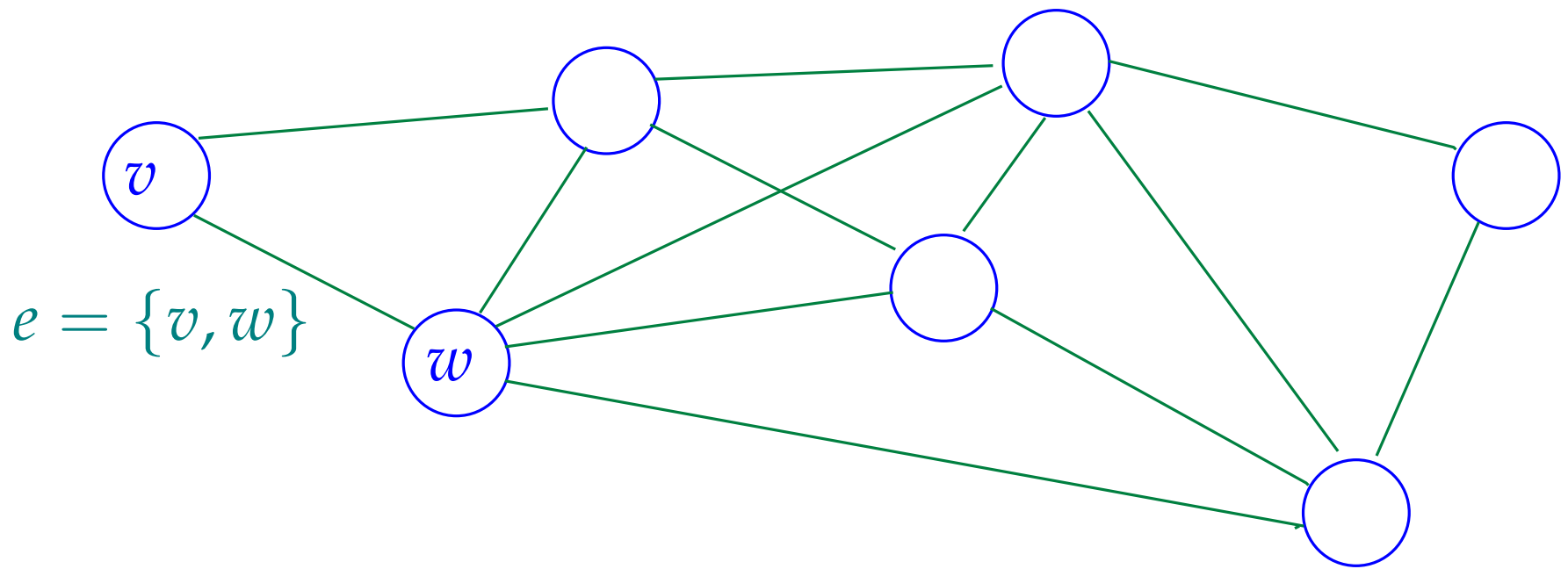
(Englisch: 'directed edge' oder 'directed arc')

# Graphen



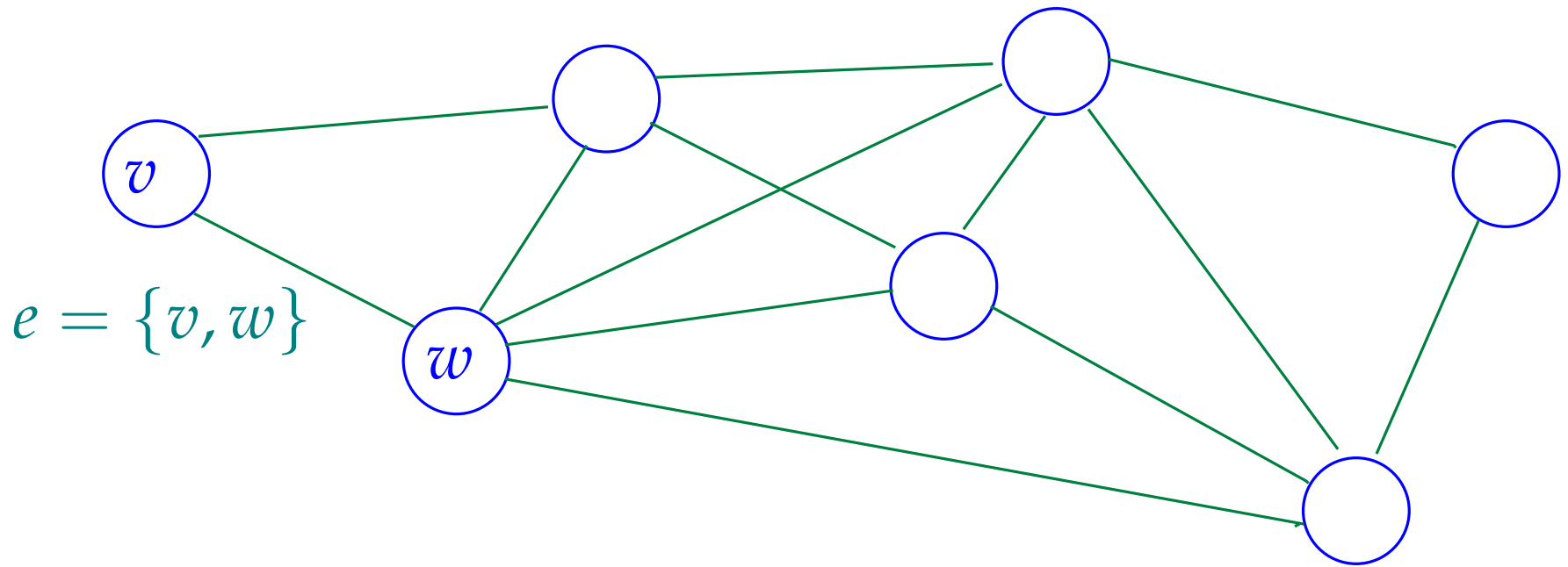
In manchen Anwendungen haben (gerichtete oder ungerichtete) Kanten **Kantenlabel**, die (beispielsweise) für Kantenlänge, Kantengewicht, Kantenkapazität, oder ähnliches stehen.

# Graphen



Zwei 'benachbarte' Knoten im Graphen heißen **adjazent**.

# Graphen

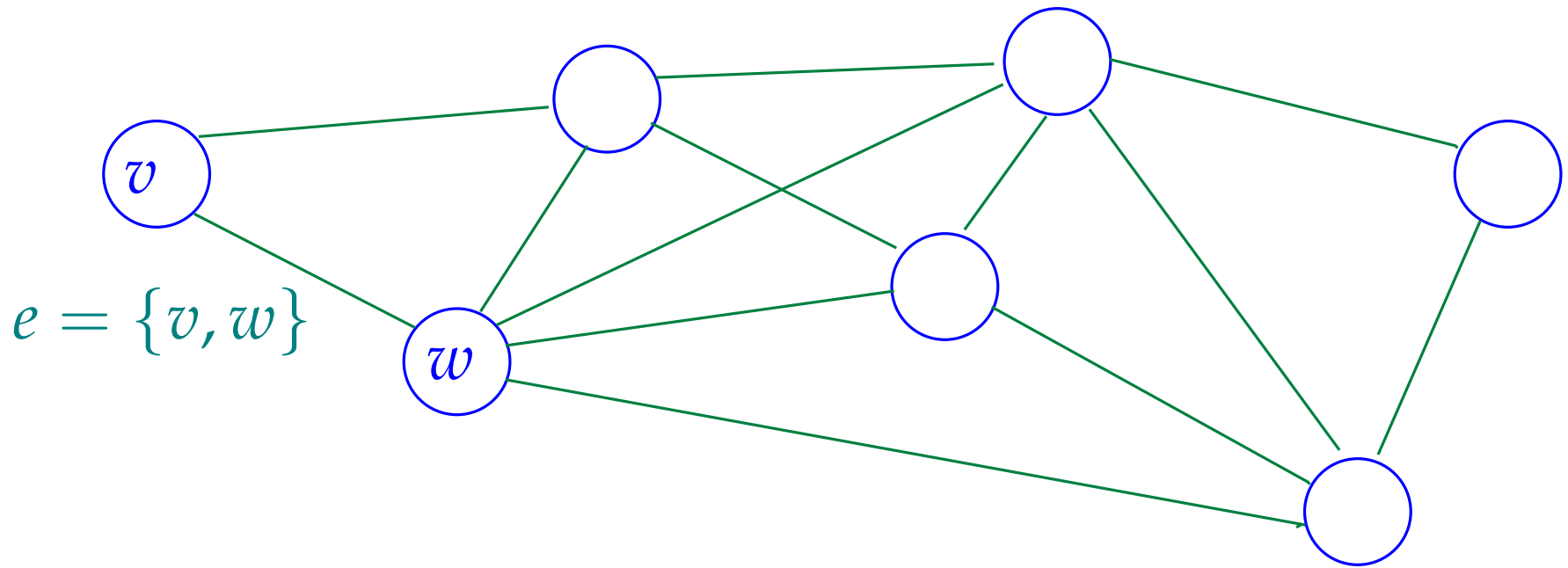


Zwei 'benachbarte' Knoten im Graphen heißen **adjazent**.

Eine Kante  $e$  und ein Knoten  $v$  heißen **inzident**, wenn der Knoten einer der Endknoten der Kante ist.



# Graphen

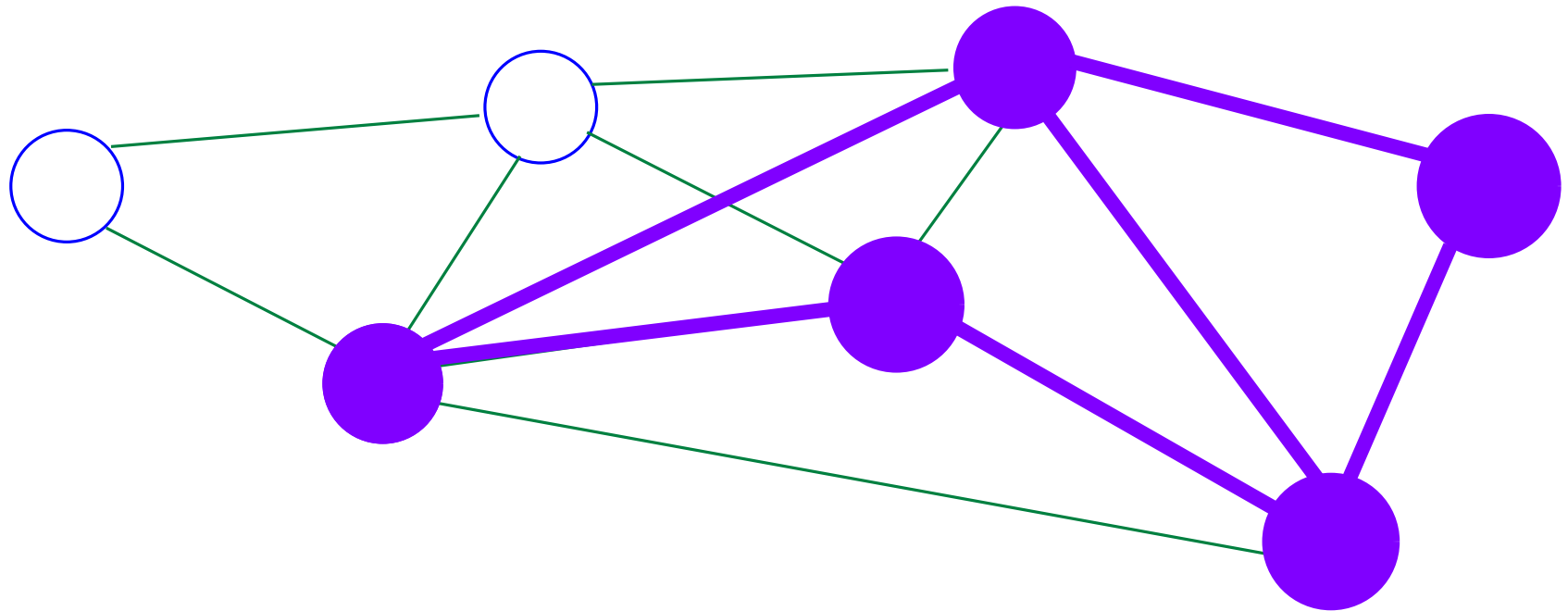


Zwei 'benachbarte' Knoten im Graphen heißen **adjazent**.

Eine Kante  $e$  und ein Knoten  $v$  heißen **inzident**, wenn der Knoten einer der Endknoten der Kante ist.

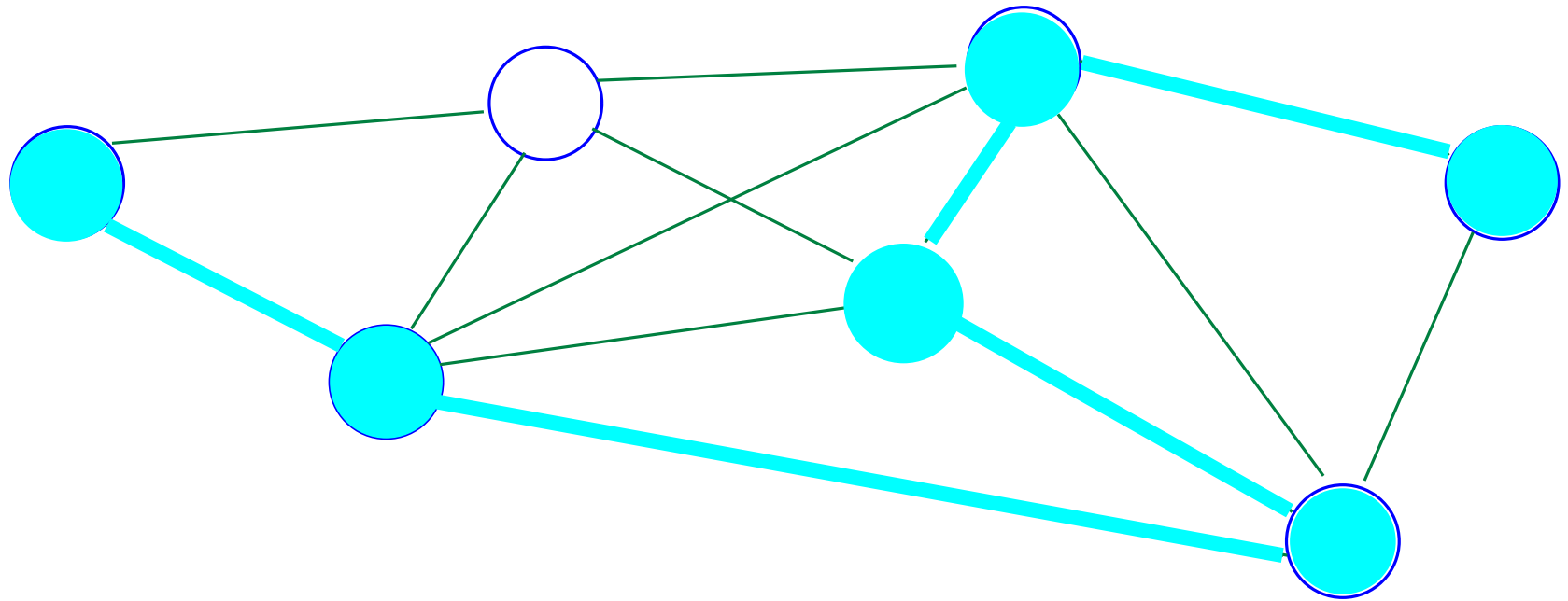
Der **Knotengrad** (Englisch: node degree)  $deg(v)$  eines Knotens  $v$  ist die Anzahl der inzidenten Kanten.

# Graphen



Ein **Subgraph**  $G' = (V', E')$  eines Graphs  $G = (V, E)$  ist ein Graph mit  $V' \subset V$  und  $E' \subset E$ .

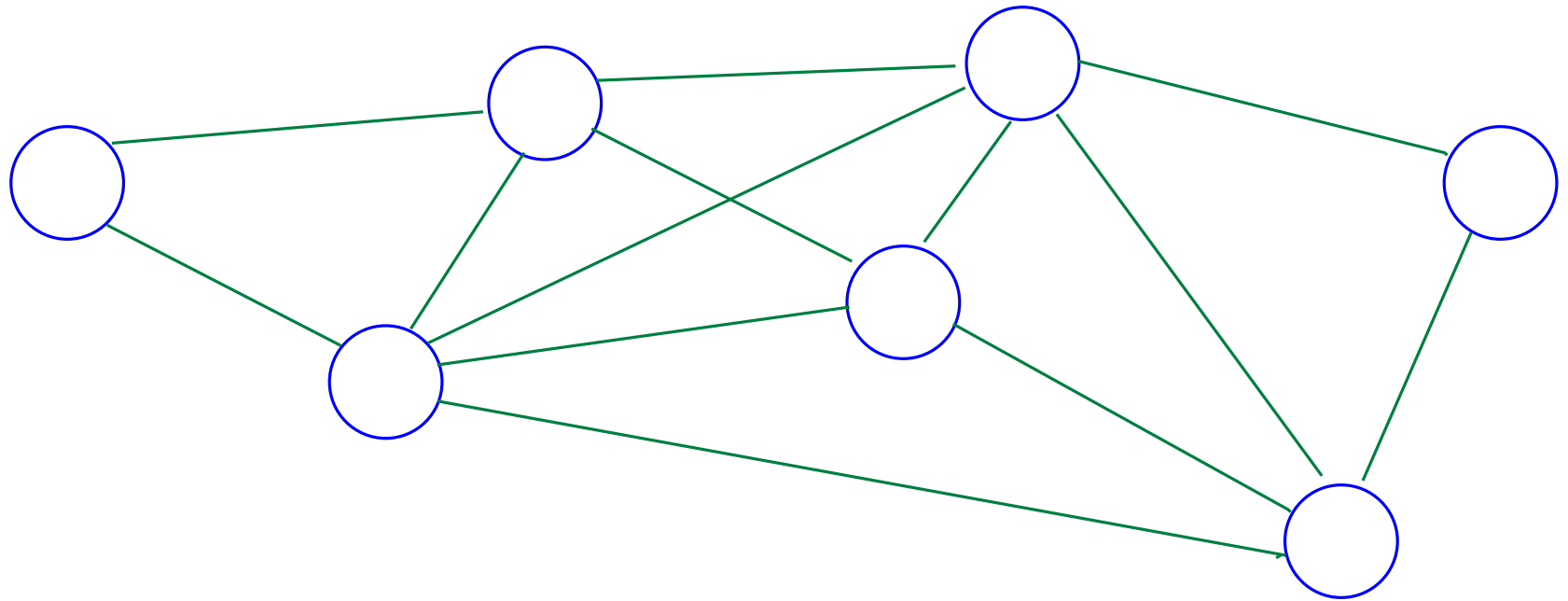
# Graphen



Ein **Weg/Pfad**  $P$  in einem Graph  $G = (V, E)$  ist eine endliche Folge  $P = (v_1, e_1, v_2, e_2, \dots, v_k)$  so dass  $e_i = \{v_i, v_{i+1}\}$  (oder  $e_i = (v_i, v_{i+1})$  in einem gerichteten Graph).

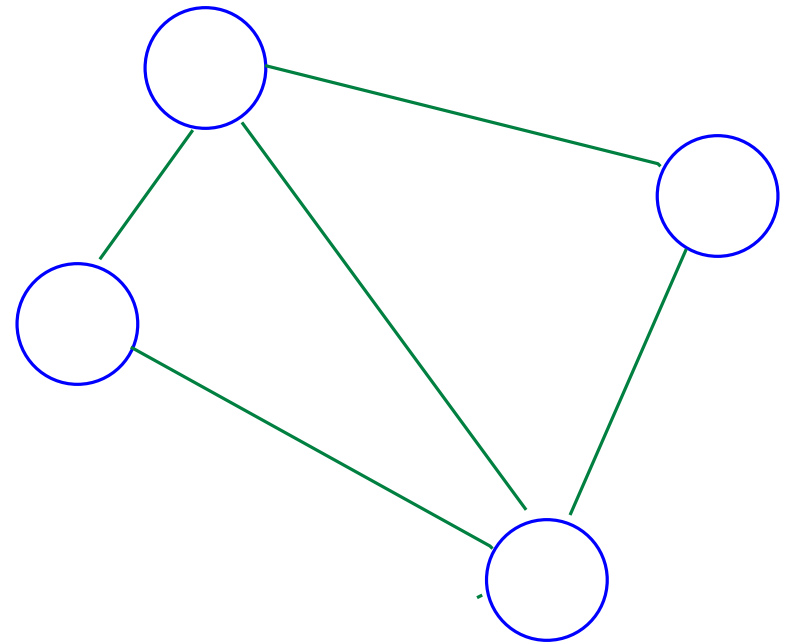
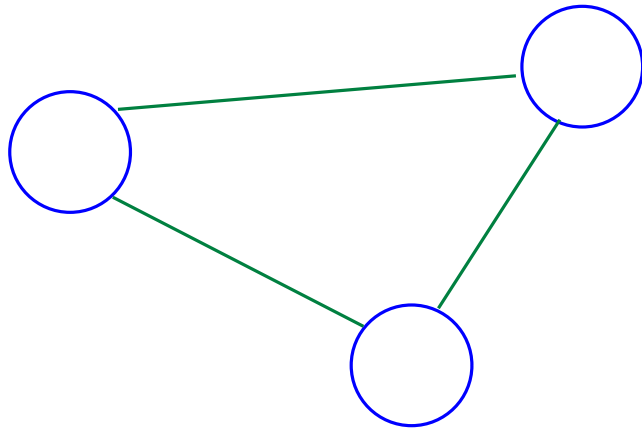
Wege werden oft auch nur als Knotensequenz oder als Kantensequenz dargestellt.

# Graphen



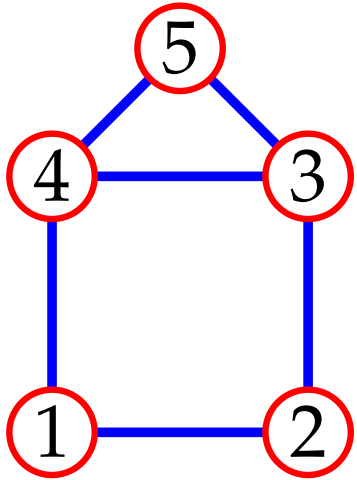
Ein Graph heißt **zusammenhängend**, wenn es zwischen jeden 2 Knoten  $v$  und  $w$  einen Weg im Graph gibt.

# Graphen



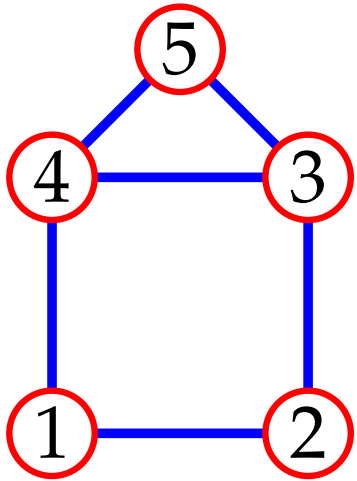
Ein Graph heißt **zusammenhängend**, wenn es zwischen jeden 2 Knoten  $v$  und  $w$  einen Weg im Graph gibt.

# F: Wie repräsentiere ich einen Graphen?

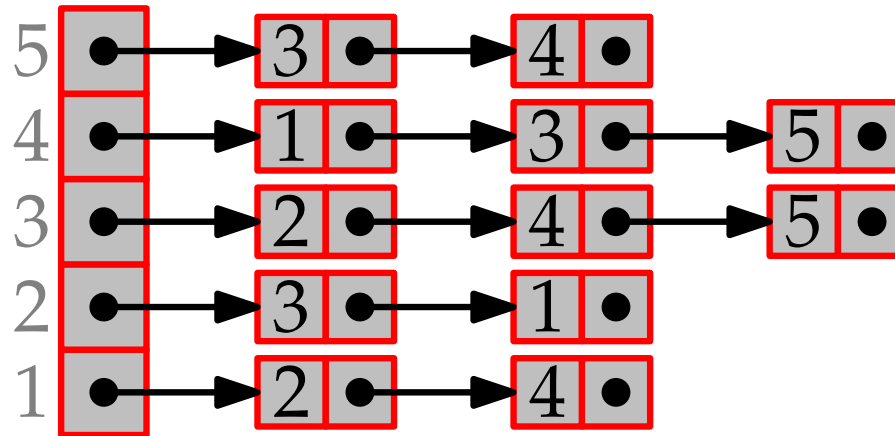


ungerichteter  
Graph

# F: Wie repräsentiere ich einen Graphen?

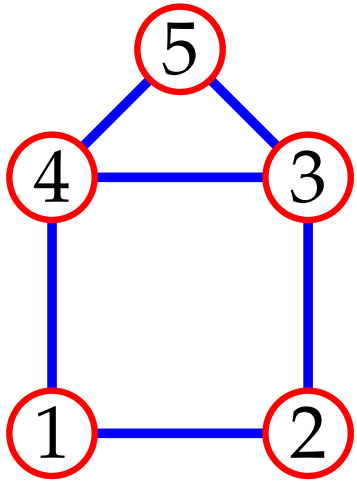


ungerichteter  
Graph

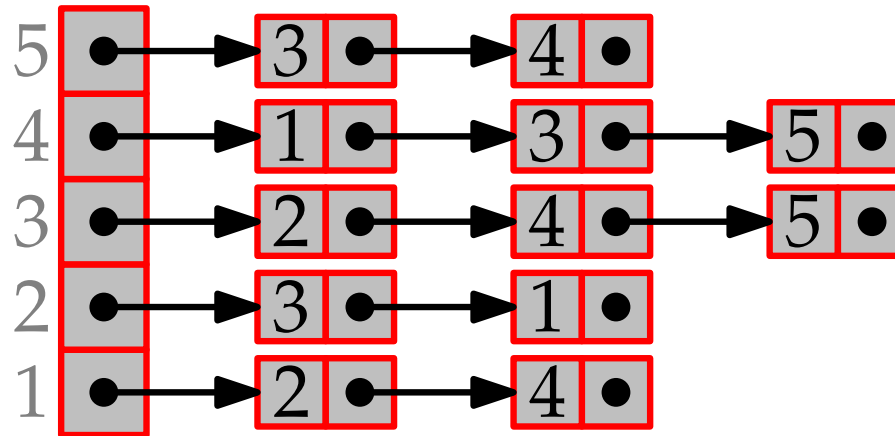


*Adjazenzlisten*

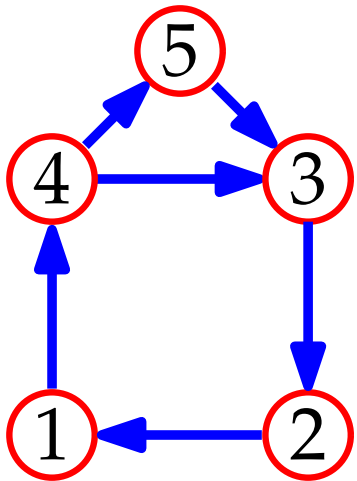
# F: Wie repräsentiere ich einen Graphen?



ungerichteter  
Graph



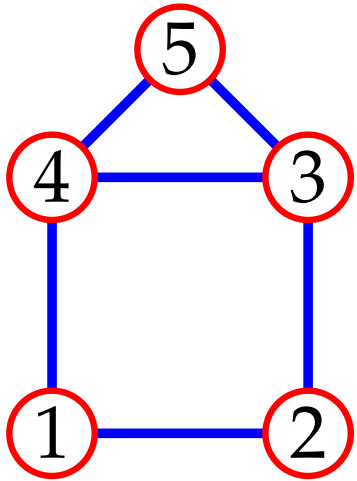
*Adjazenzlisten*



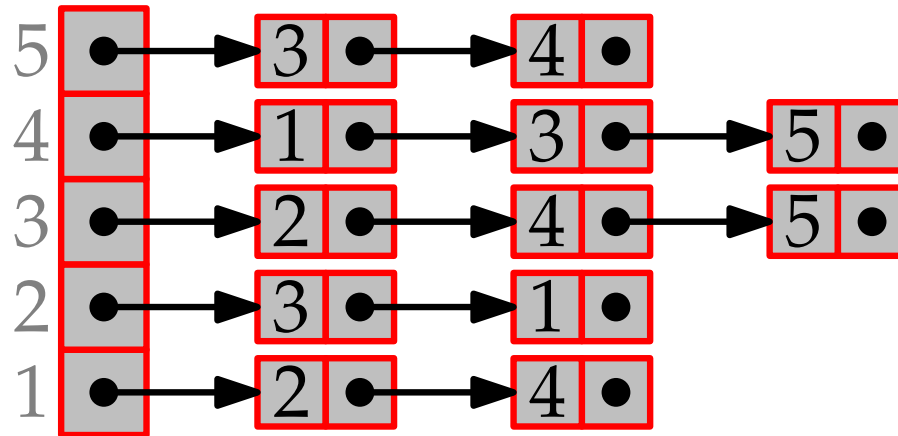
gerichteter  
Graph



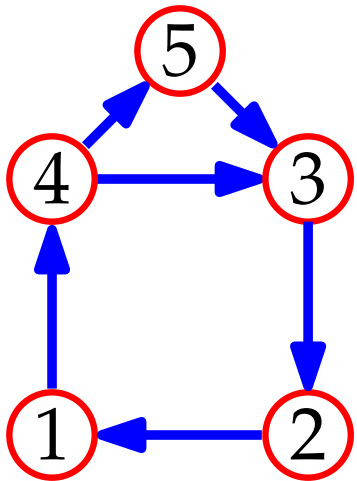
# F: Wie repräsentiere ich einen Graphen?



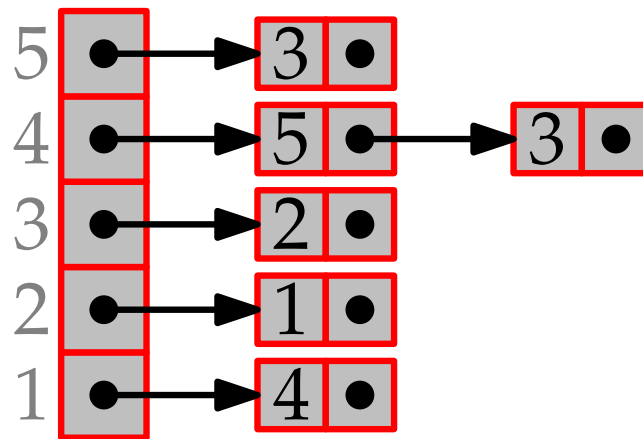
ungerichteter  
Graph



*Adjazenzlisten*

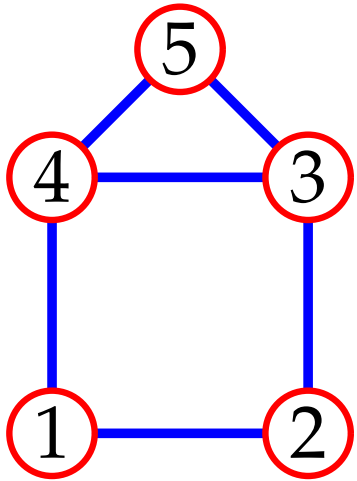


gerichteter  
Graph

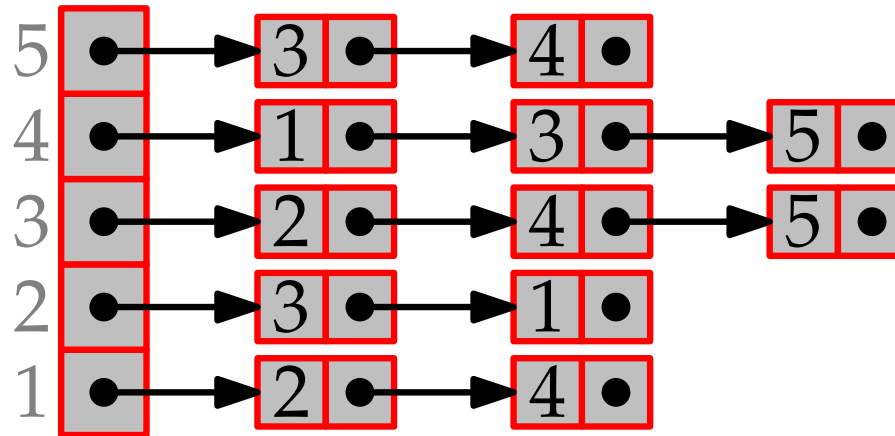


$$\text{Adj}[i] = \{j \in V(G) : (i, j) \in E(G)\}$$

# F: Wie repräsentiere ich einen Graphen?



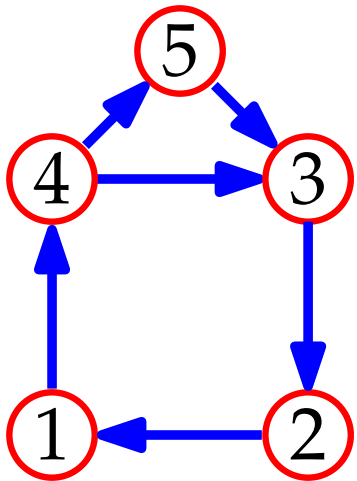
ungerichteter Graph



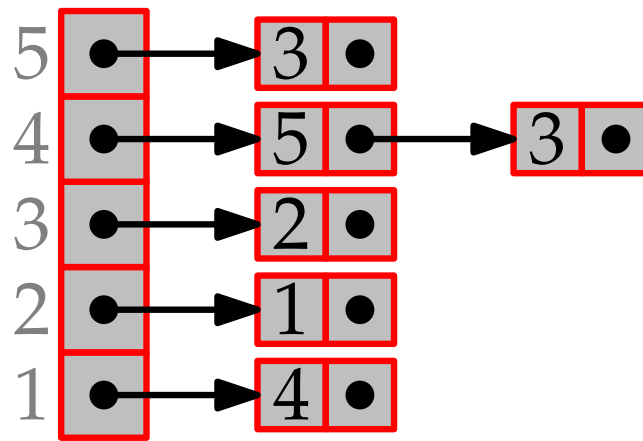
*Adjazenzlisten*

	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*

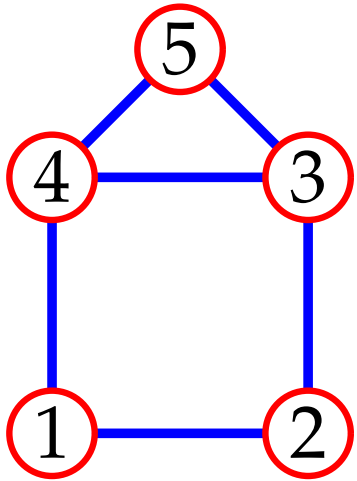


gerichteter Graph

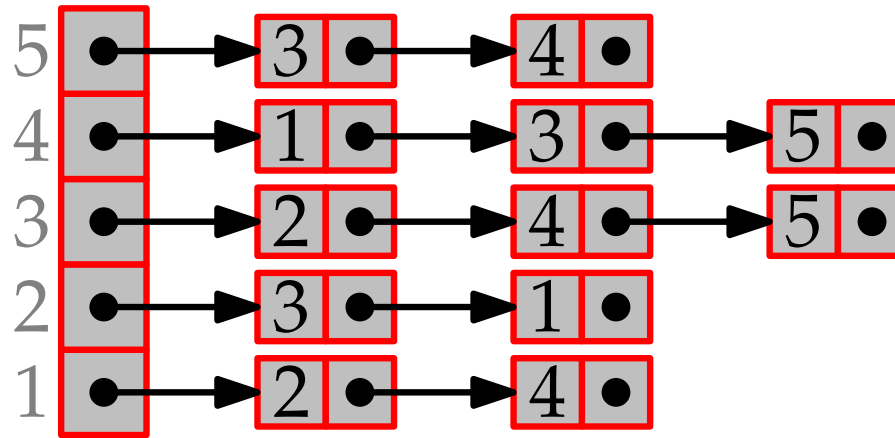


$$\text{Adj}[i] = \{j \in V(G) : (i, j) \in E(G)\}$$

# F: Wie repräsentiere ich einen Graphen?



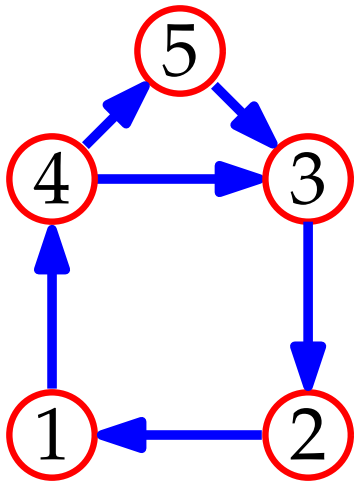
ungerichteter Graph



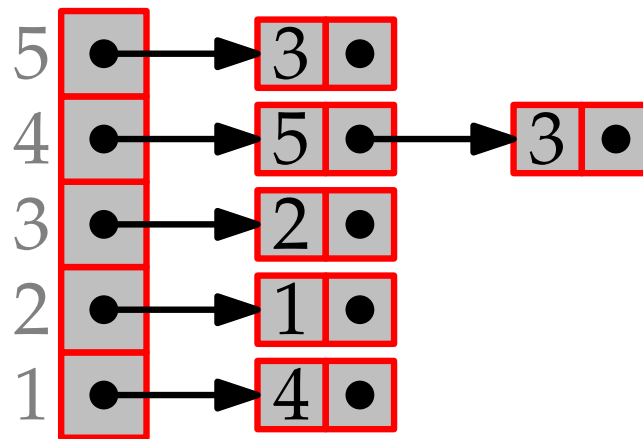
*Adjazenzlisten*

	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*



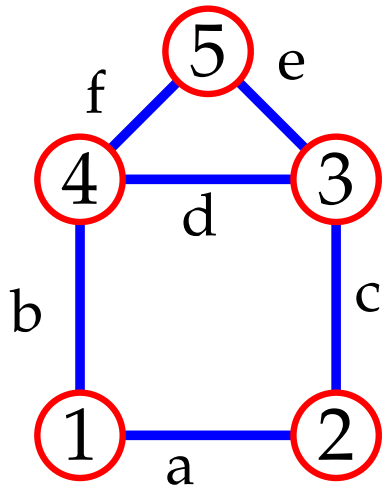
gerichteter Graph



$$\text{Adj}[i] = \{j \in V(G) : (i, j) \in E(G)\} \quad a_{ij} = 1 \Leftrightarrow (i, j) \in E(G)$$

	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

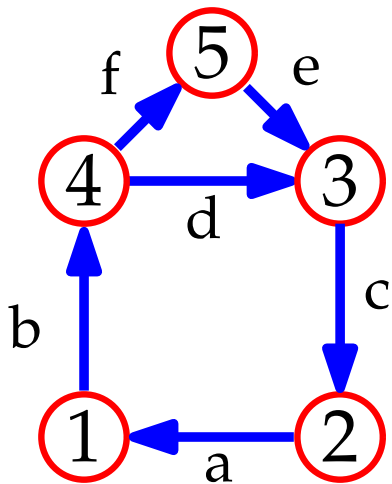
# F: Wie repräsentiere ich einen Graphen?



ungerichteter  
Graph

	a	b	c	d	e	f
1	1	1	0	0	0	0
2	1	0	1	0	0	0
3	0	0	1	1	1	0
4	0	1	0	1	0	1
5	0	0	0	0	1	1

*Knoten-Kanten-Inzidenzmatrix*

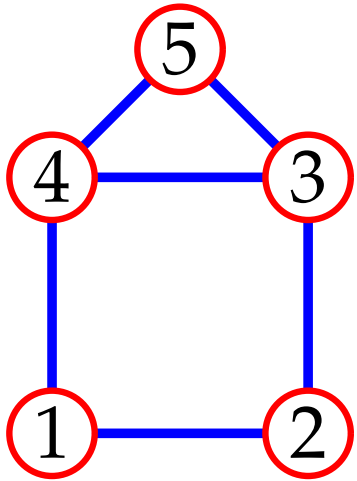


gerichteter  
Graph

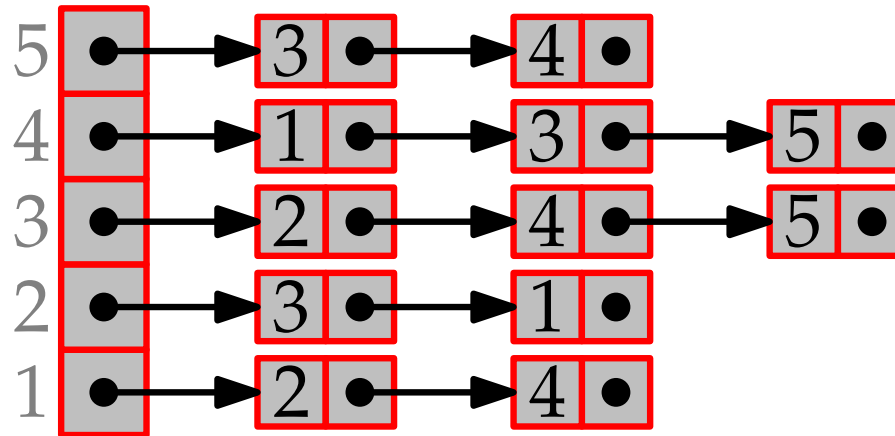
	a	b	c	d	e	f
1	-1	1	0	0	0	0
2	1	0	-1	0	0	0
3	0	0	1	-1	-1	0
4	0	-1	0	1	0	1
5	0	0	0	0	1	-1

*Knoten-Kanten-Inzidenzmatrix*

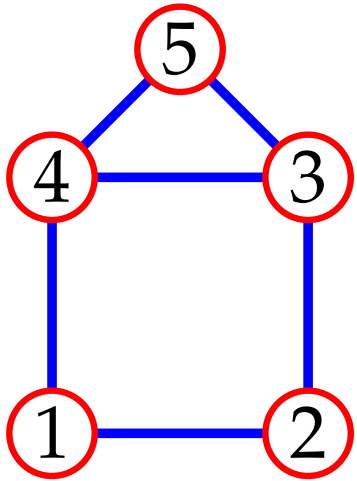
# F: Wie repräsentiere ich einen Graphen?



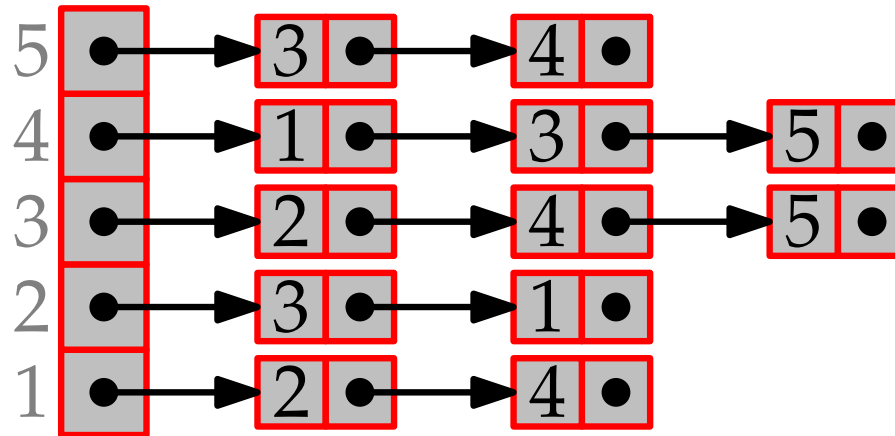
ungerichteter  
Graph



# F: Wie repräsentiere ich einen Graphen?

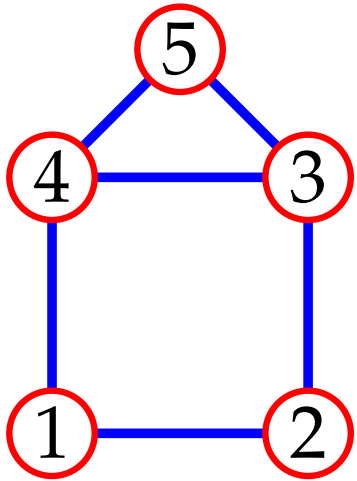


ungerichteter  
Graph

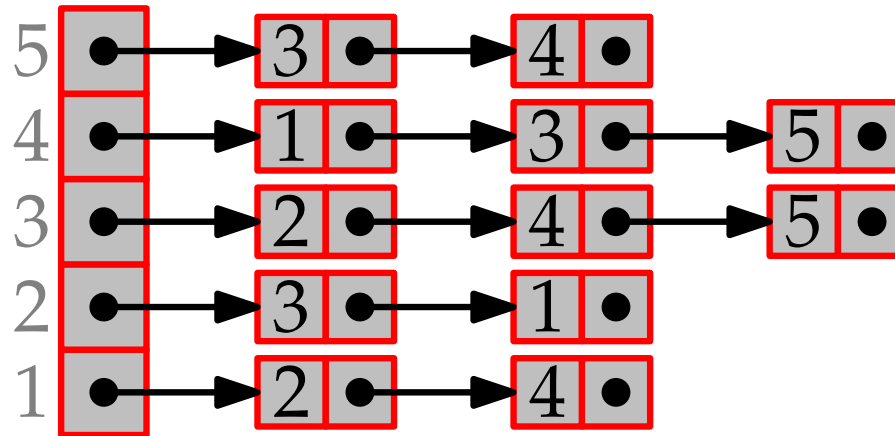


*Adjazenzlisten*

# F: Wie repräsentiere ich einen Graphen?

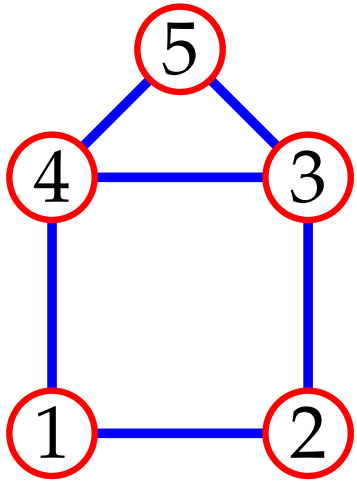


ungerichteter  
Graph

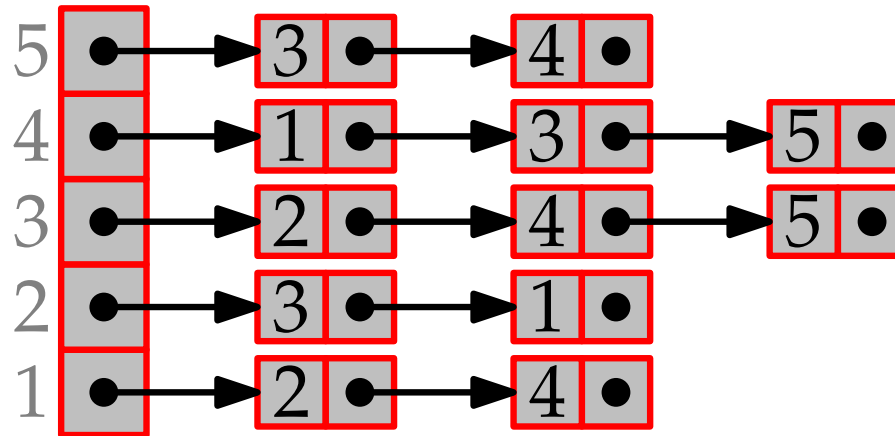


*Adjazenzlisten*

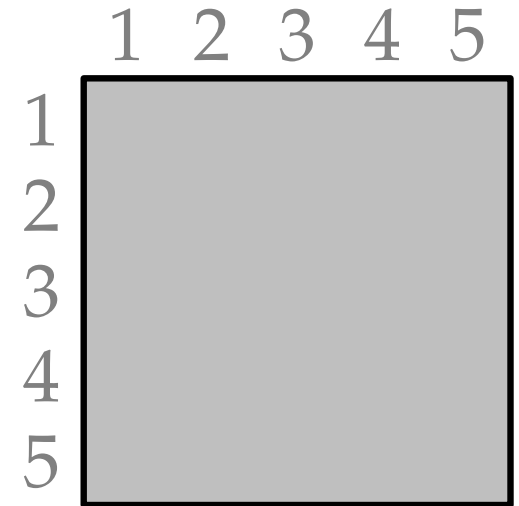
# F: Wie repräsentiere ich einen Graphen?



ungerichteter  
Graph

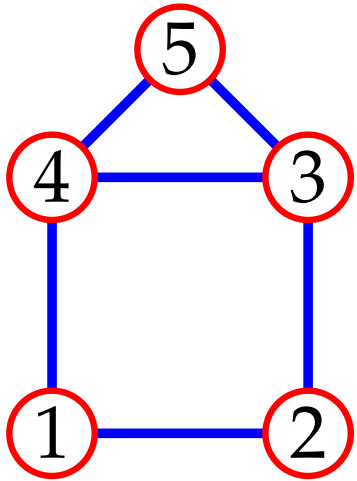


*Adjazenzlisten*

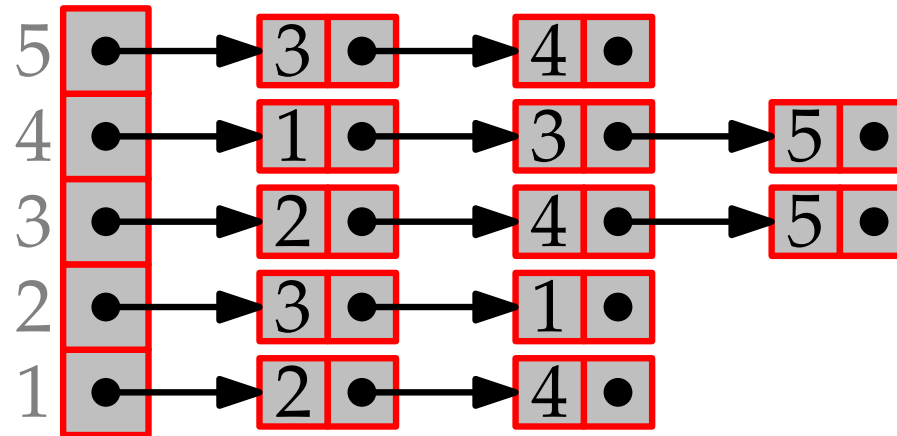




# F: Wie repräsentiere ich einen Graphen?



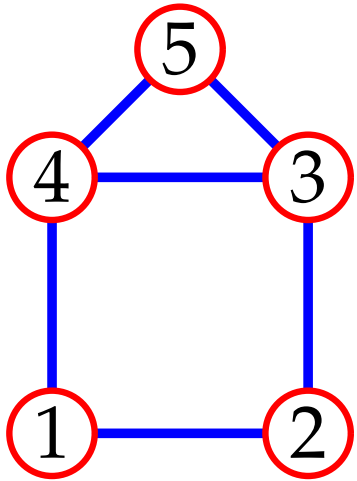
ungerichteter  
Graph



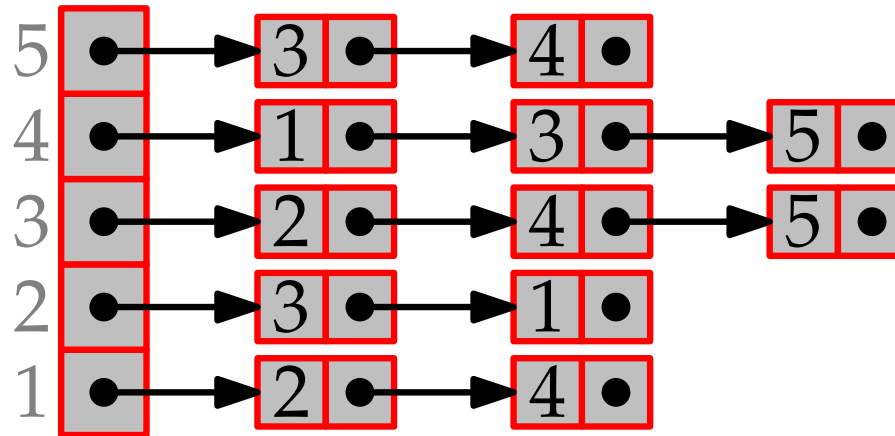
*Adjazenzlisten*

	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

# F: Wie repräsentiere ich einen Graphen?



ungerichteter  
Graph

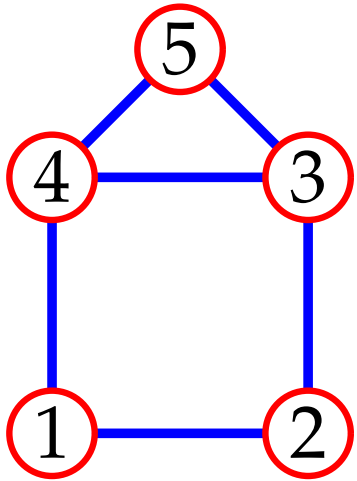


*Adjazenzlisten*

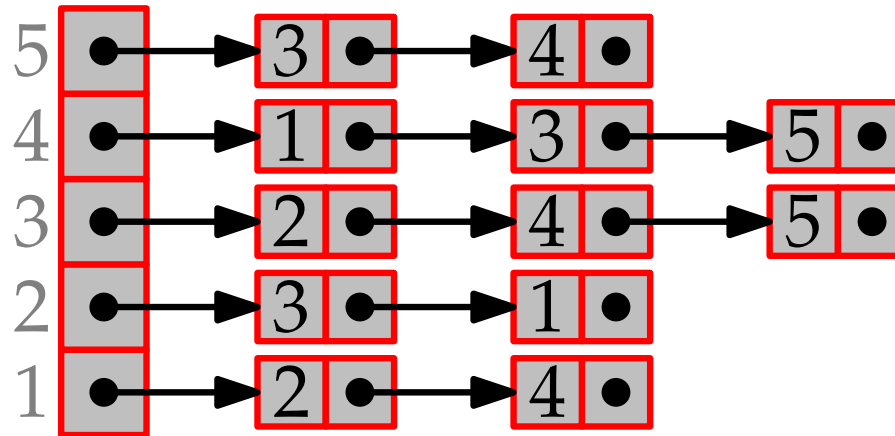
	1	2	3	4	5
1	0	1	0	1	0
2	1	0	1	0	0
3	0	1	0	1	1
4	1	0	1	0	1
5	0	0	1	1	0

*Adjazenzmatrix*

# F: Wie repräsentiere ich einen Graphen?



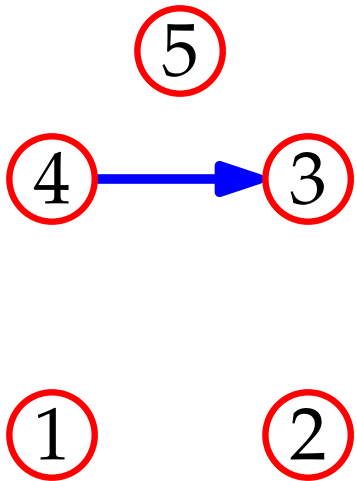
ungerichteter  
Graph



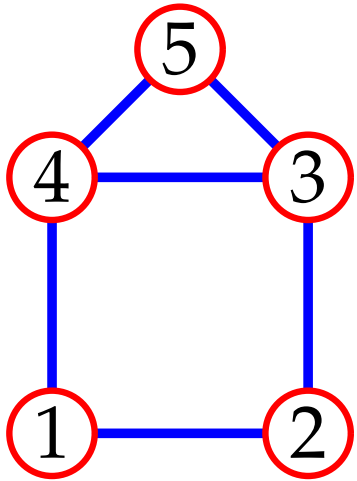
*Adjazenzlisten*

	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

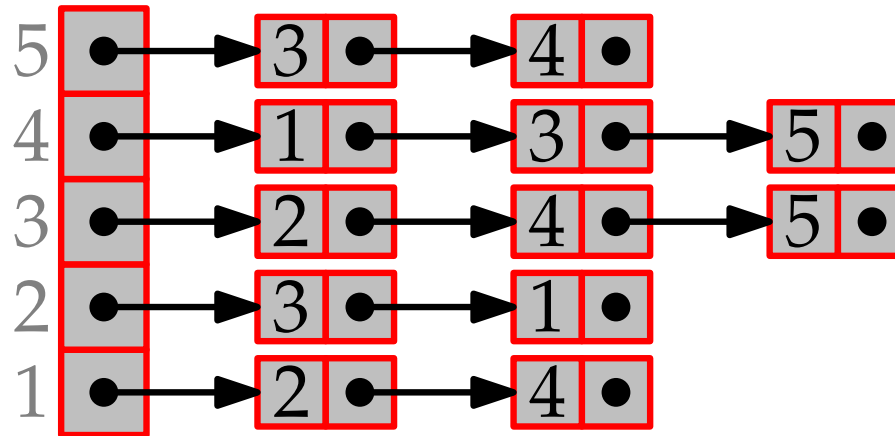
*Adjazenzmatrix*



# F: Wie repräsentiere ich einen Graphen?



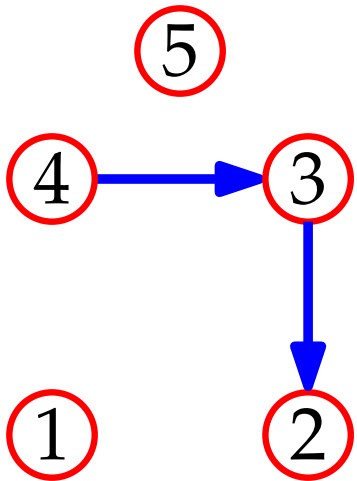
ungerichteter  
Graph



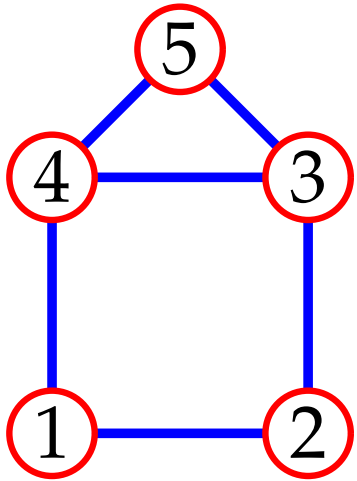
*Adjazenzlisten*

	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

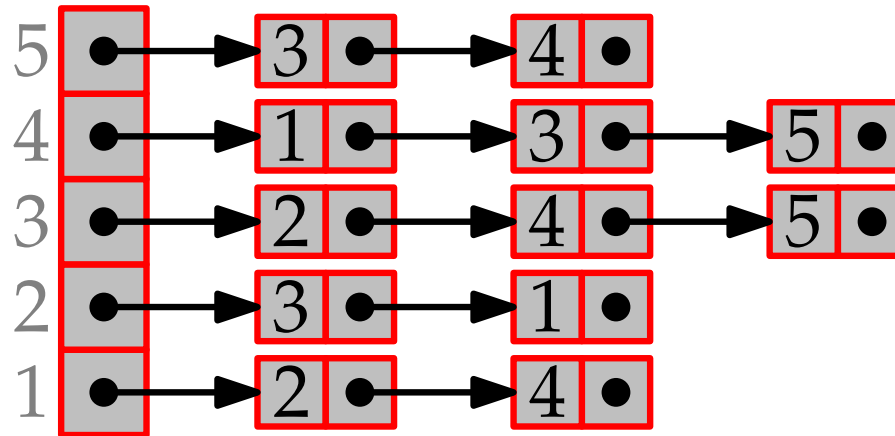
*Adjazenzmatrix*



# F: Wie repräsentiere ich einen Graphen?



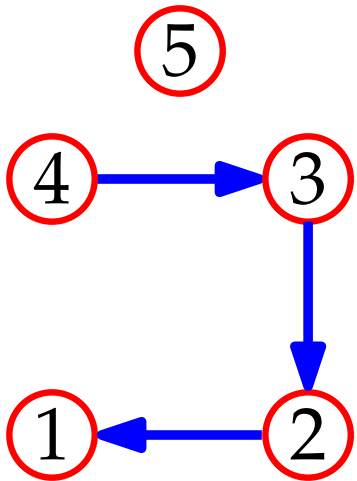
ungerichteter  
Graph



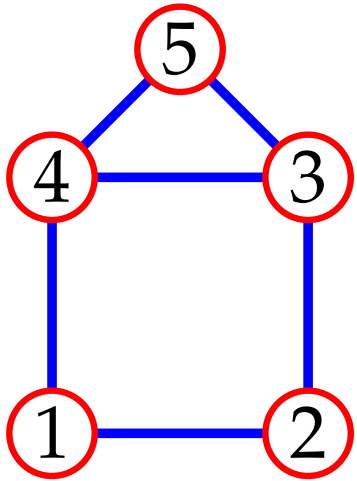
*Adjazenzlisten*

	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

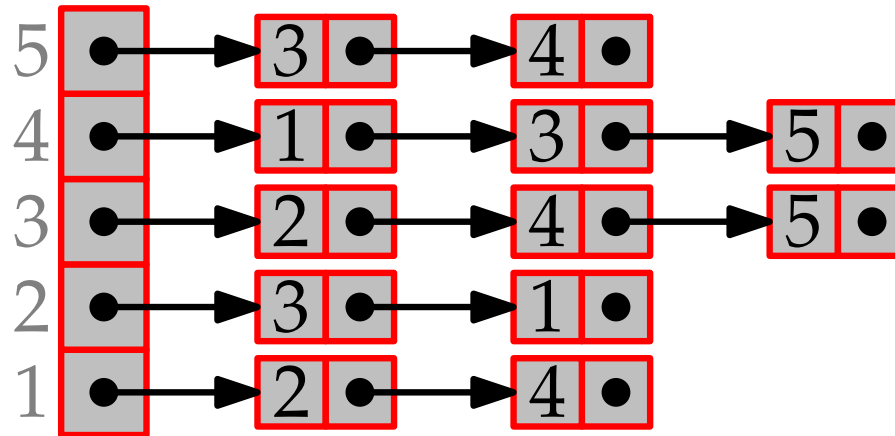
*Adjazenzmatrix*



# F: Wie repräsentiere ich einen Graphen?



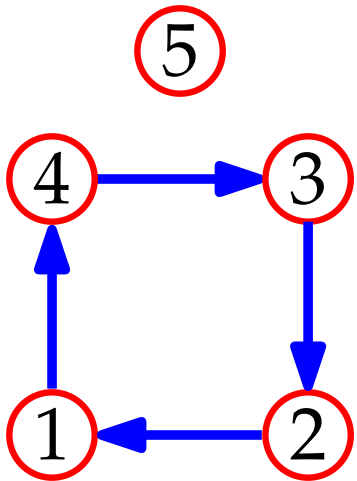
ungerichteter  
Graph



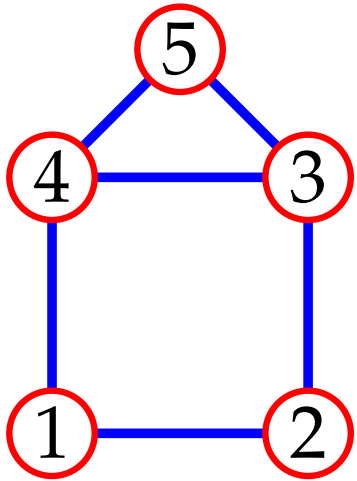
*Adjazenzlisten*

	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

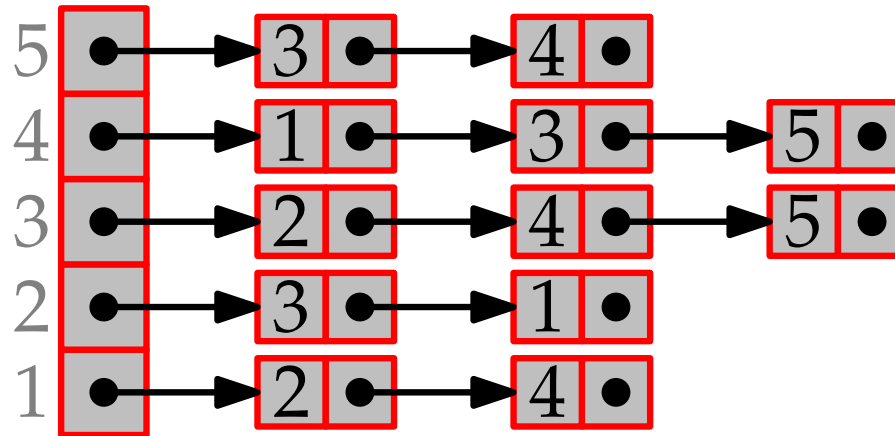
*Adjazenzmatrix*



# F: Wie repräsentiere ich einen Graphen?



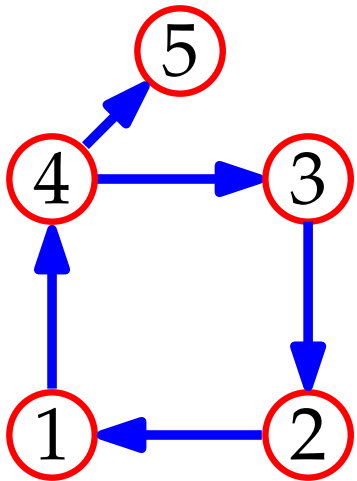
ungerichteter  
Graph



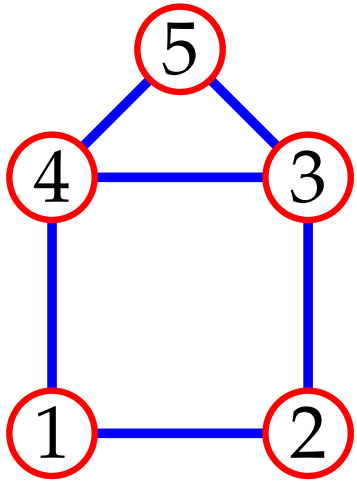
*Adjazenzlisten*

	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

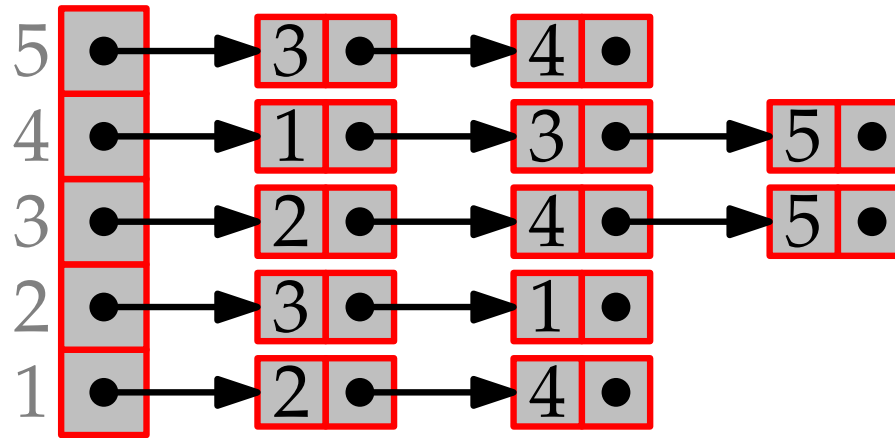
*Adjazenzmatrix*



# F: Wie repräsentiere ich einen Graphen?



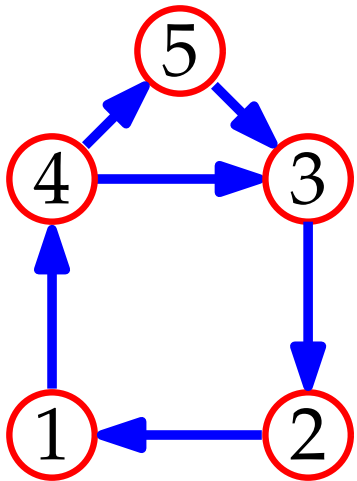
ungerichteter  
Graph



*Adjazenzlisten*

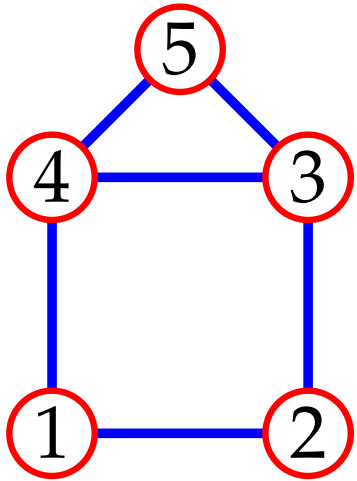
	1	2	3	4	5
1	0	1	0	1	0
2	1	0	1	0	0
3	0	1	0	1	1
4	1	0	1	0	1
5	0	0	1	1	0

*Adjazenzmatrix*

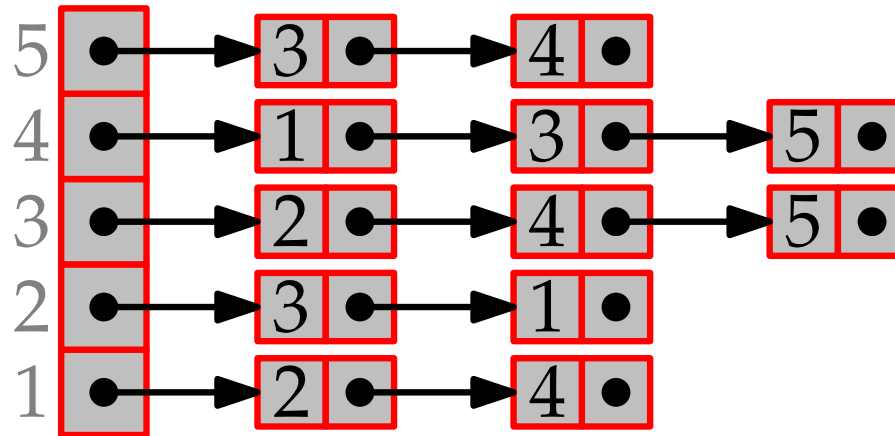




# F: Wie repräsentiere ich einen Graphen?



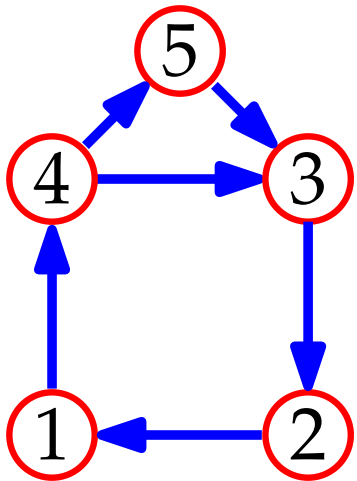
ungerichteter  
Graph



*Adjazenzlisten*

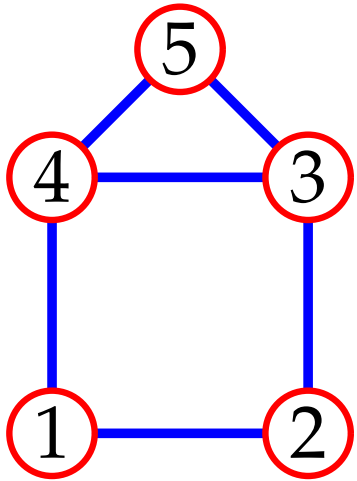
	1	2	3	4	5
1	0	1	0	1	0
2	1	0	1	0	0
3	0	1	0	1	1
4	1	0	1	0	1
5	0	0	1	1	0

*Adjazenzmatrix*

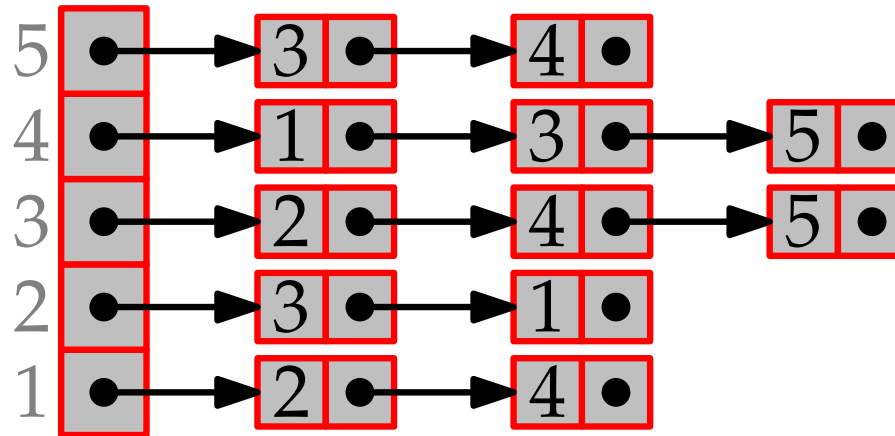


gerichteter  
Graph

# F: Wie repräsentiere ich einen Graphen?



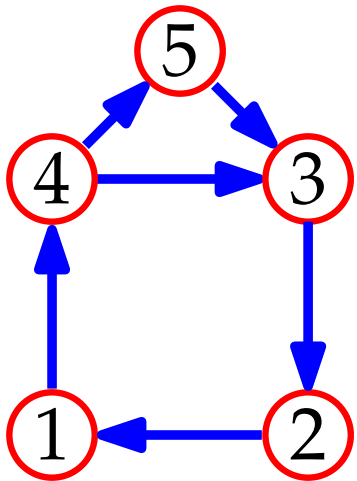
ungerichteter Graph



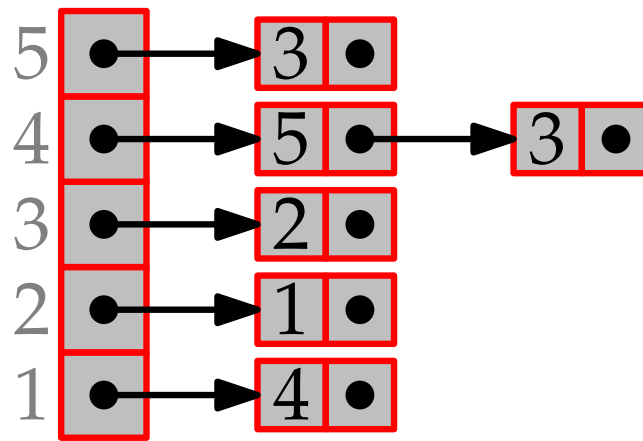
*Adjazenzlisten*

	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*

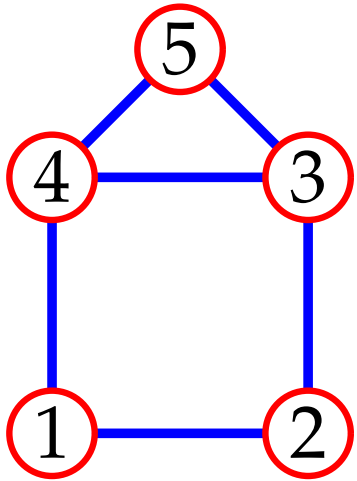


gerichteter Graph

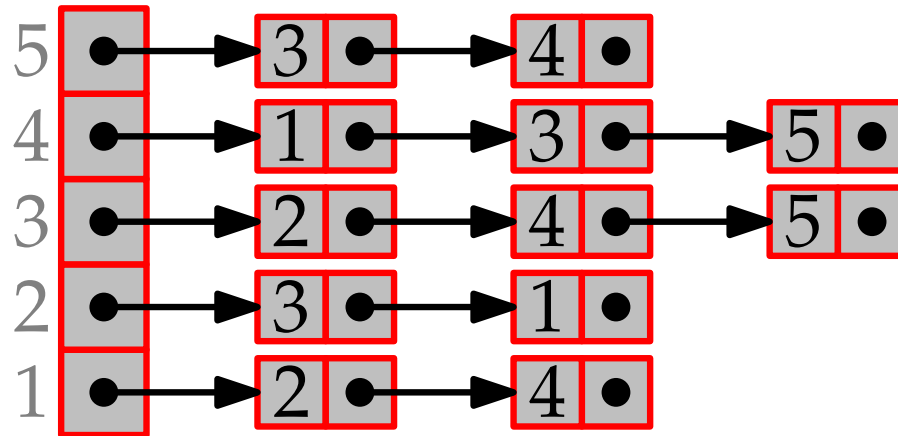


$$\text{Adj}[i] = \{j \in V(G) : (i, j) \in E(G)\}$$

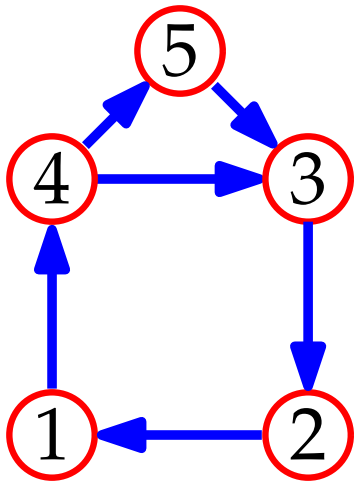
# F: Wie repräsentiere ich einen Graphen?



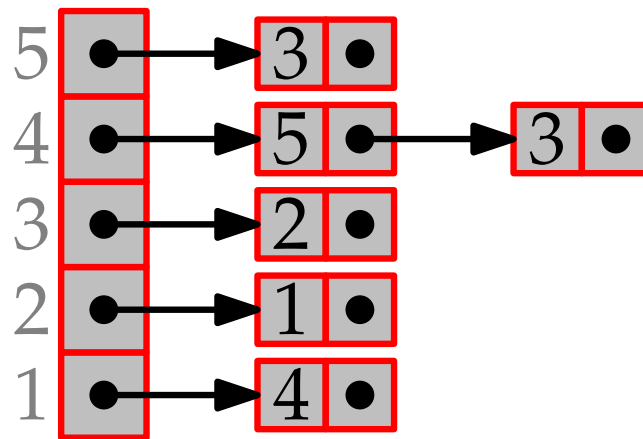
ungerichteter  
Graph



*Adjazenzlisten*

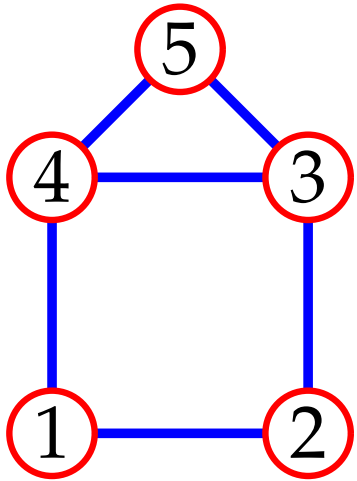


gerichteter  
Graph

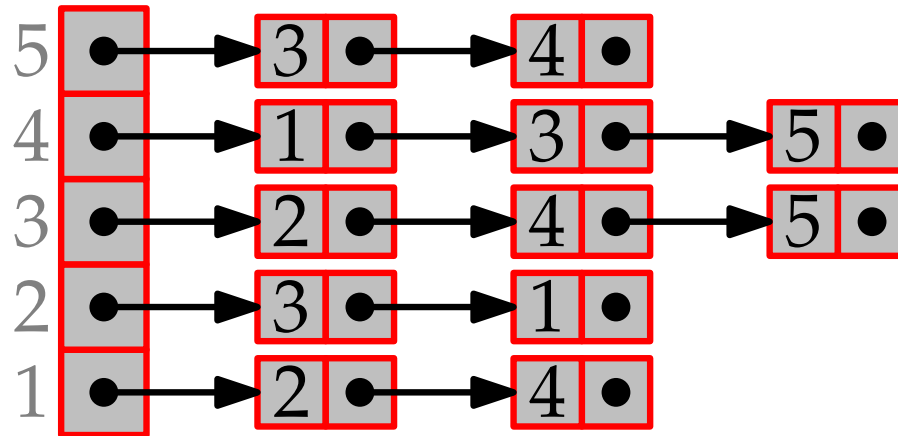


$$\text{Adj}[i] = \{j \in V(G) : (i, j) \in E(G)\}$$

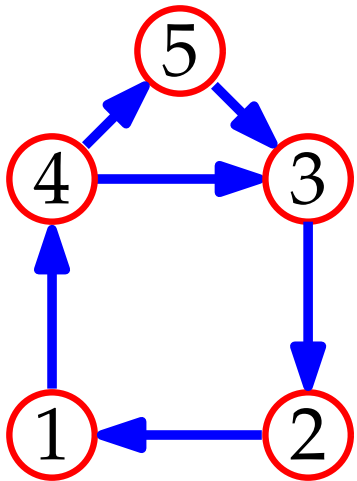
# F: Wie repräsentiere ich einen Graphen?



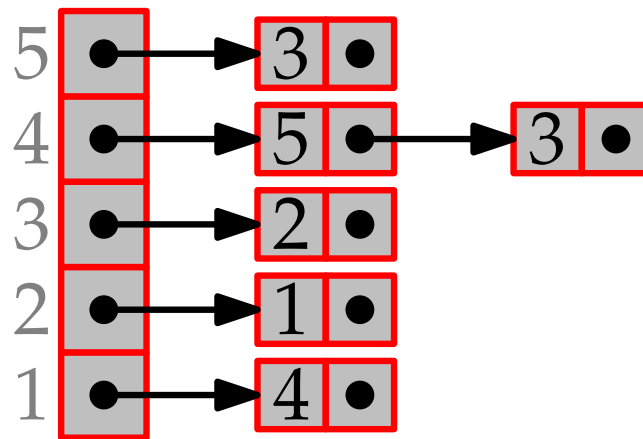
ungerichteter  
Graph



*Adjazenzlisten*



gerichteter  
Graph



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

# Agenda

1. Worum es hier geht ✓
2. Formales und Organisatorisches ✓
3. Graphen - erste Definitionen und Konzepte ✓
4. **Modellieren mit Graphen**

# Modellieren mit Graphen

Modellieren bedeutet:

# Modellieren mit Graphen

Modellieren bedeutet:

'Übersetzen'

# Modellieren mit Graphen

Modellieren bedeutet:

'Übersetzen'

im Kontext (von Teil 1) dieser Vorlesung meist: aus der realen Welt  
in ein Optimierungsproblem auf einem Graph



# Modellieren mit Graphen

Modellieren bedeutet:

'Übersetzen'

im Kontext (von Teil 1) dieser Vorlesung meist: aus der realen Welt  
in ein Optimierungsproblem auf einem Graph

'auf das Wesentliche reduzieren'

# Modellieren mit Graphen

Modellieren bedeutet:

'Übersetzen'

im Kontext (von Teil 1) dieser Vorlesung meist: aus der realen Welt  
in ein Optimierungsproblem auf einem Graph

'auf das Wesentliche reduzieren'

Hier hat man **Spielraum** als Modellierer und es gibt normalerweise  
nicht die *eine* richtige Antwort.

# Modellieren mit Graphen

Modellieren bedeutet:

'Übersetzen'

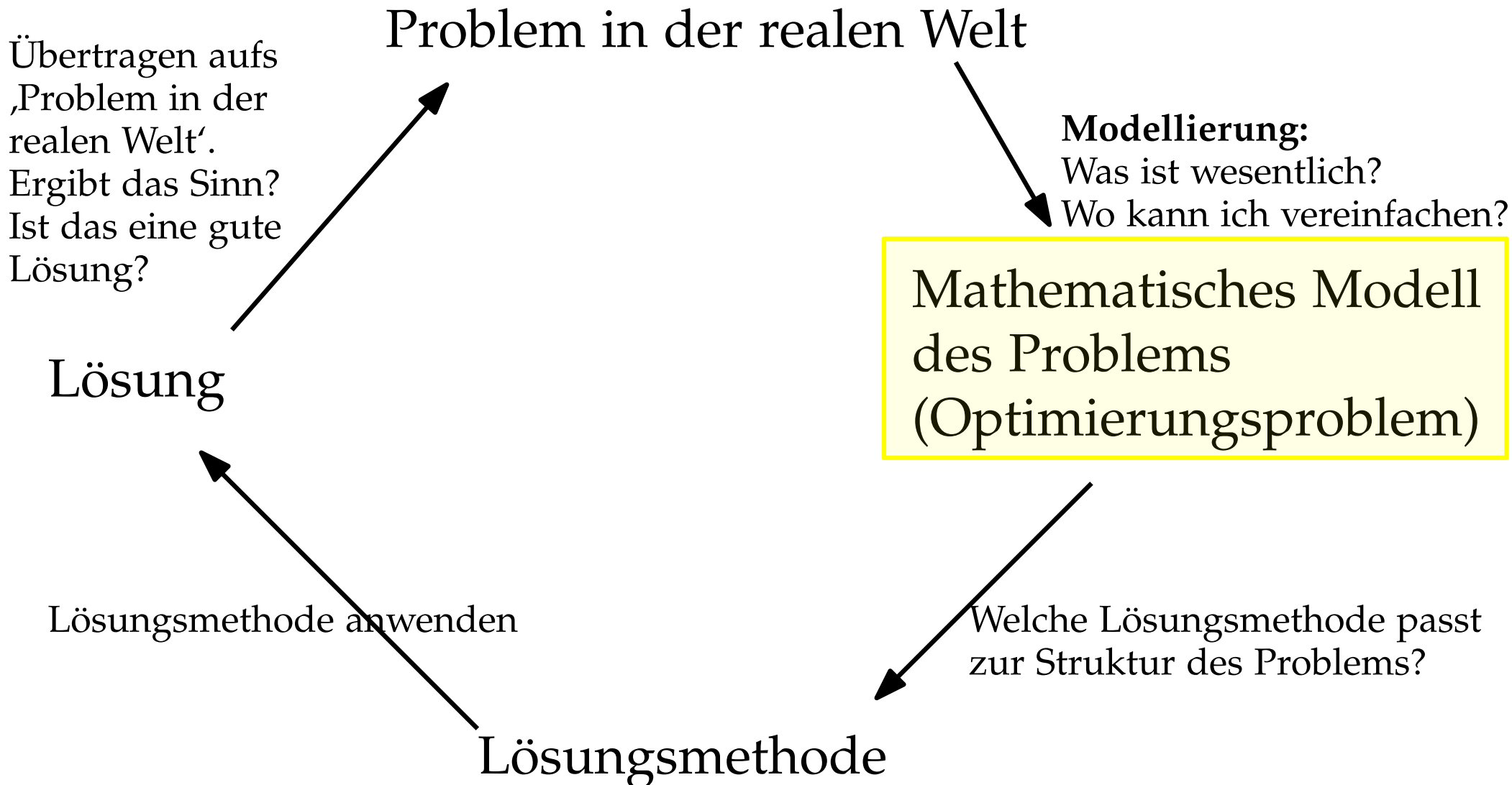
im Kontext (von Teil 1) dieser Vorlesung meist: aus der realen Welt in ein Optimierungsproblem auf einem Graph

'auf das Wesentliche reduzieren'

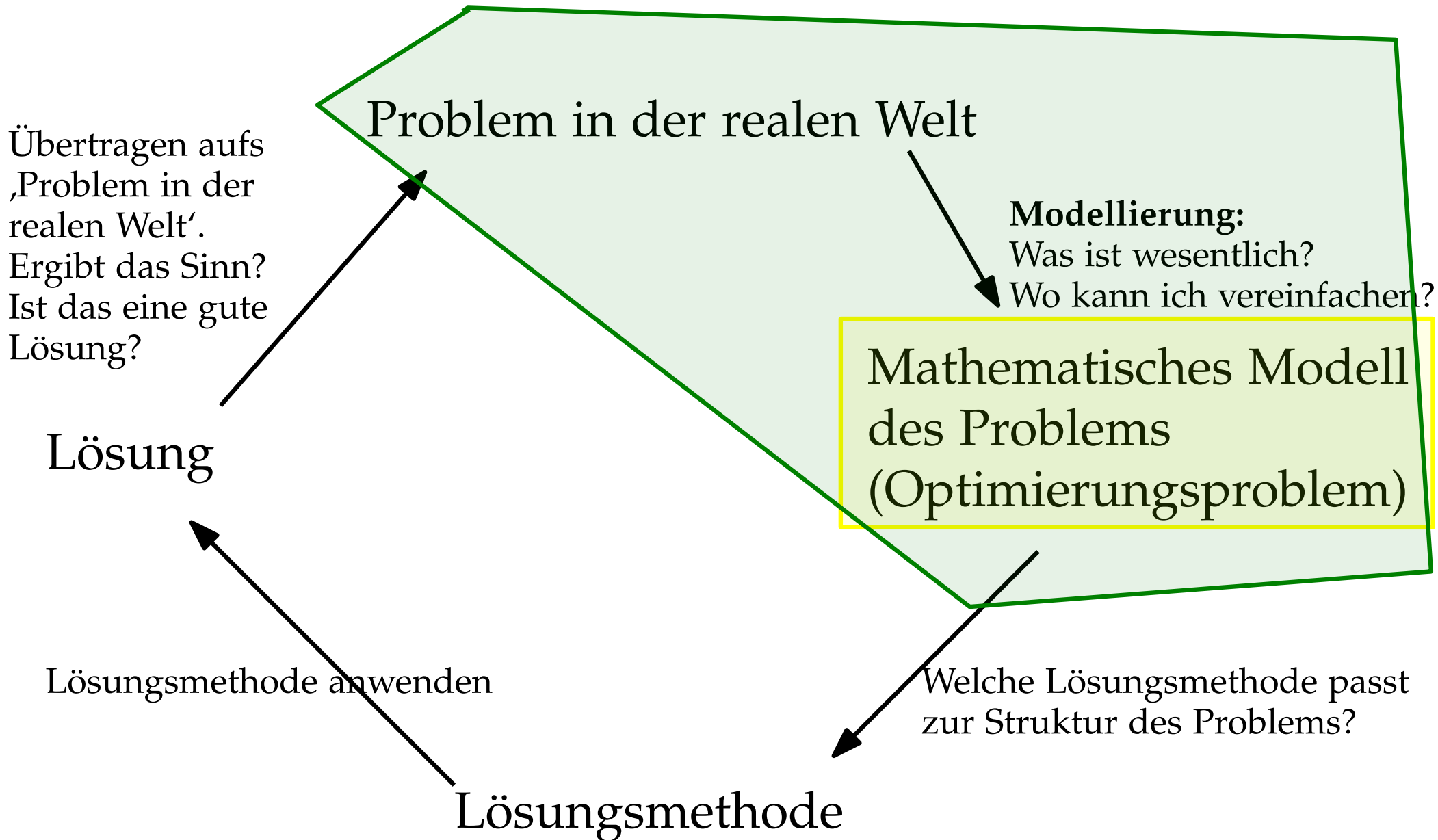
Hier hat man **Spielraum** als Modellierer und es gibt normalerweise nicht die *eine* richtige Antwort.

Im Verlauf eines Projekts muss diese Frage oft mehrmals aufgegriffen werden.

# Modellierungszyklus in der Optimierung



# Modellierungszyklus in der Optimierung



# Optimierungsproblem - Zutaten

- Gegeben: . . . (Probleminstanz)
- Gesucht: . . . (zulässiger Bereich, Zielfunktion, 'max' oder 'min')

**Beispiel:** kürzeste Wege-Problem

# Optimierungsproblem - Zutaten

- Gegeben: . . . (Probleminstanz)
- Gesucht: . . . (zulässiger Bereich, Zielfunktion, 'max' oder 'min')

**Beispiel: Clique**

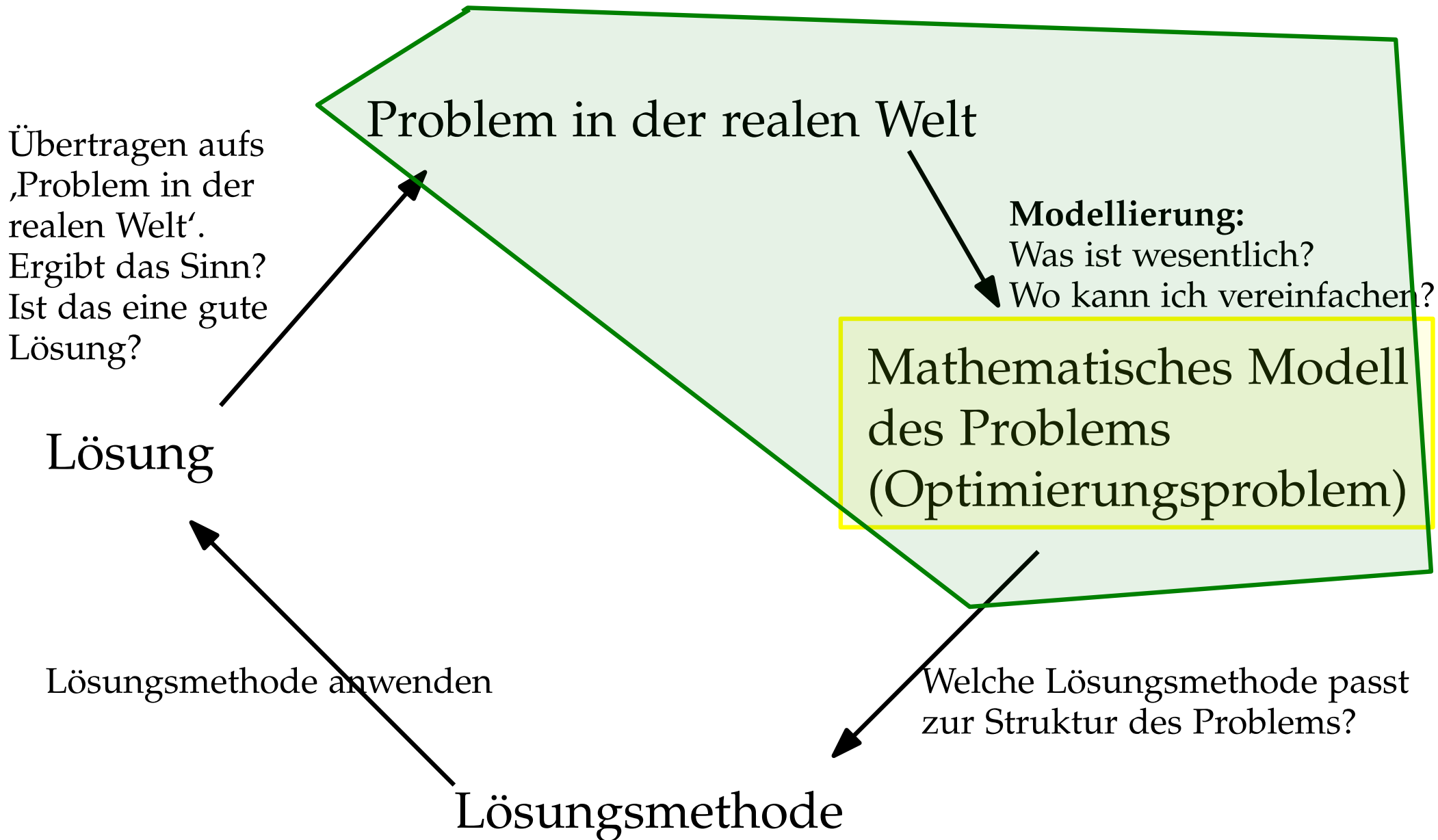
# Optimierungsproblem - Zutaten

- Gegeben: . . . (Probleminstanz)
- Gesucht: . . . (zulässiger Bereich, Zielfunktion, 'max' oder 'min')

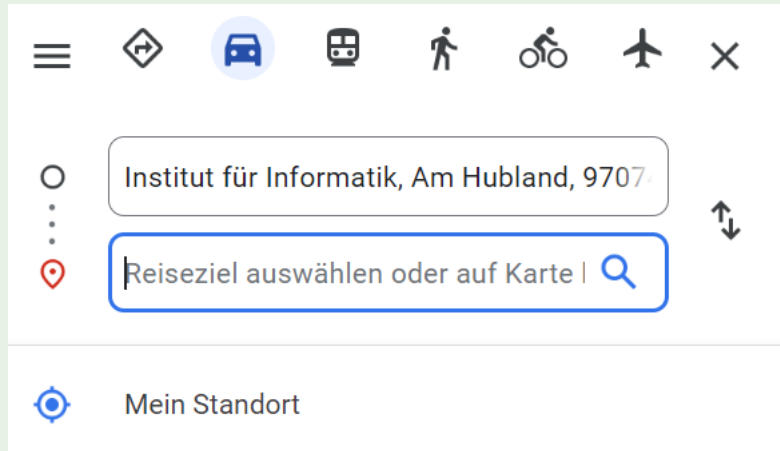
**Beispiel:** kürzeste Wege-Problem



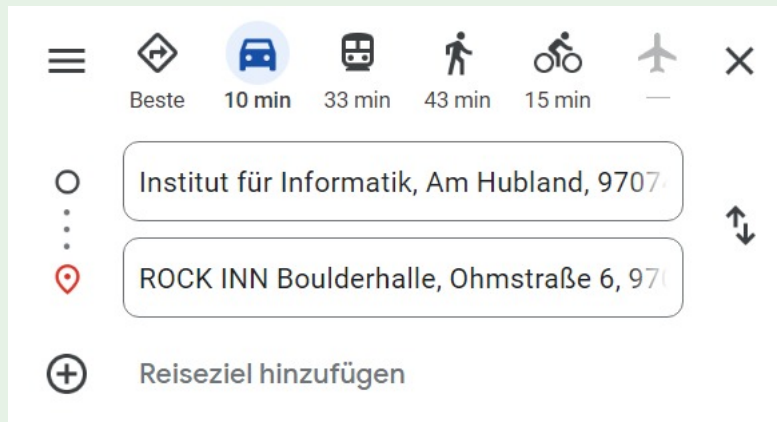
# Modellierungszyklus in der Optimierung



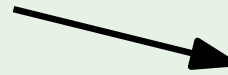
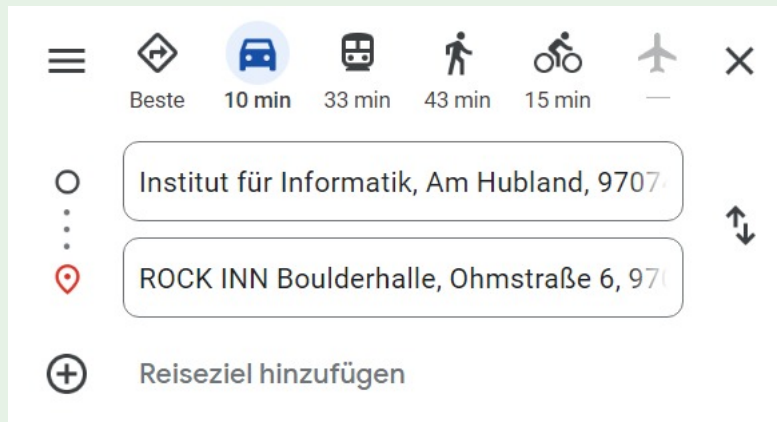
# Beispiel zur Modellierung in Graphen 1



# Beispiel zur Modellierung in Graphen 1



# Beispiel zur Modellierung in Graphen 1



← von Institut für Informatik, Am Hubland, 97074 Würzburg  
nach ROCK INN Boulderhalle, Ohmstraße 6, 97076 Würzburg

**10 min (4,5 km)**

über Am Galgenberg

Die aktuell schnellste Route aufgrund der Verkehrslage

---

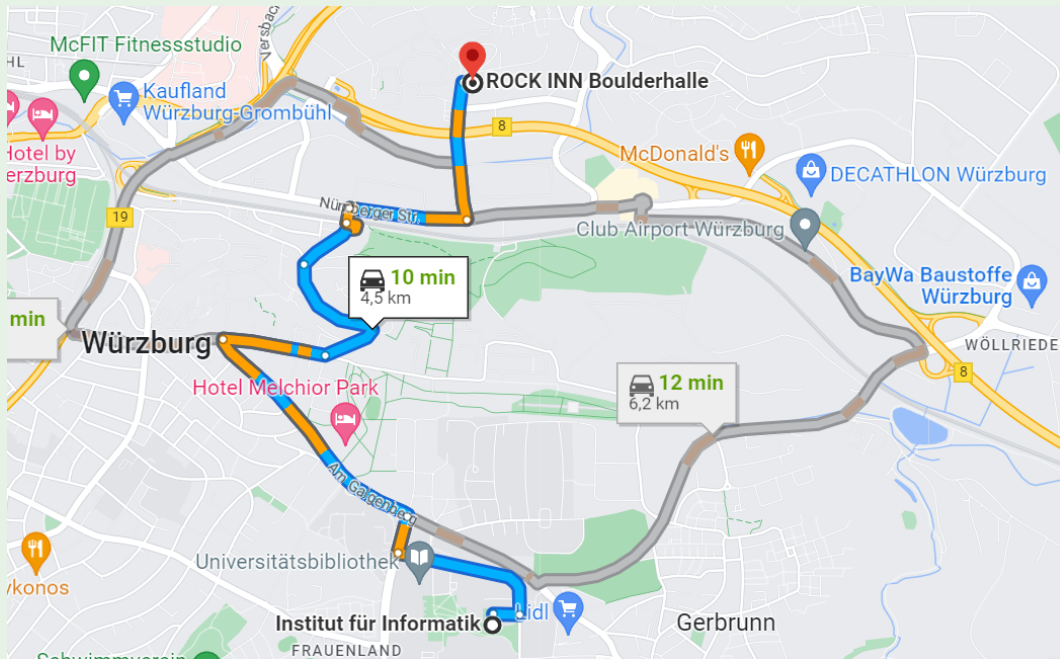
**Institut für Informatik**  
Am Hubland, 97074 Würzburg

- Theodor-Boveri-Weg bis Am Hubland nehmen  
2 min (750 m)
- Am Galgenberg bis Rottendorfer Str. nehmen  
3 min (1,2 km)
- Rottendorfer Str. und Zweierweg bis Nürnberger Str. folgen  
4 min (1,5 km)
- Ohmstraße folgen  
2 min (1,0 km)

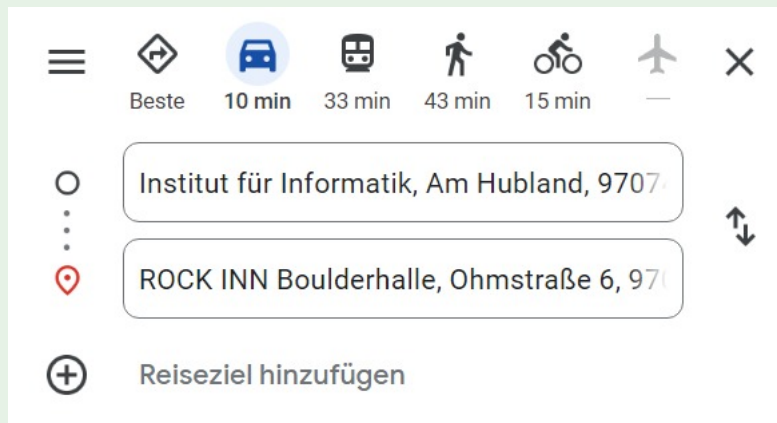
↪ Rechts abbiegen  
**i** Das Ziel befindet sich auf der linken Seite.  
12 Sek. (23 m)

---




**ROCK INN Boulderhalle**  
Ohmstraße 6, 97076 Würzburg



# Beispiel zur Modellierung in Graphen 1




← von Institut für Informatik, Am Hubland, 97074 Würzburg  
nach ROCK INN Boulderhalle, Ohmstraße 6, 97076 Würzburg

**10 min (4,5 km)**   

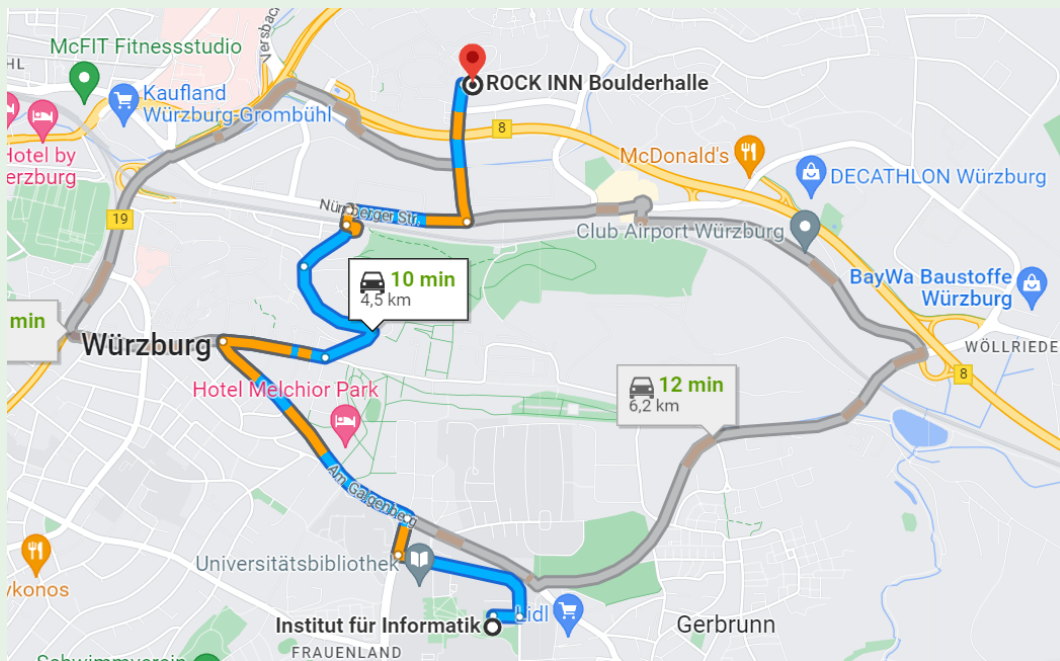
über Am Galgenberg  
Die aktuell schnellste Route aufgrund der Verkehrslage

**Institut für Informatik**  
Am Hubland, 97074 Würzburg

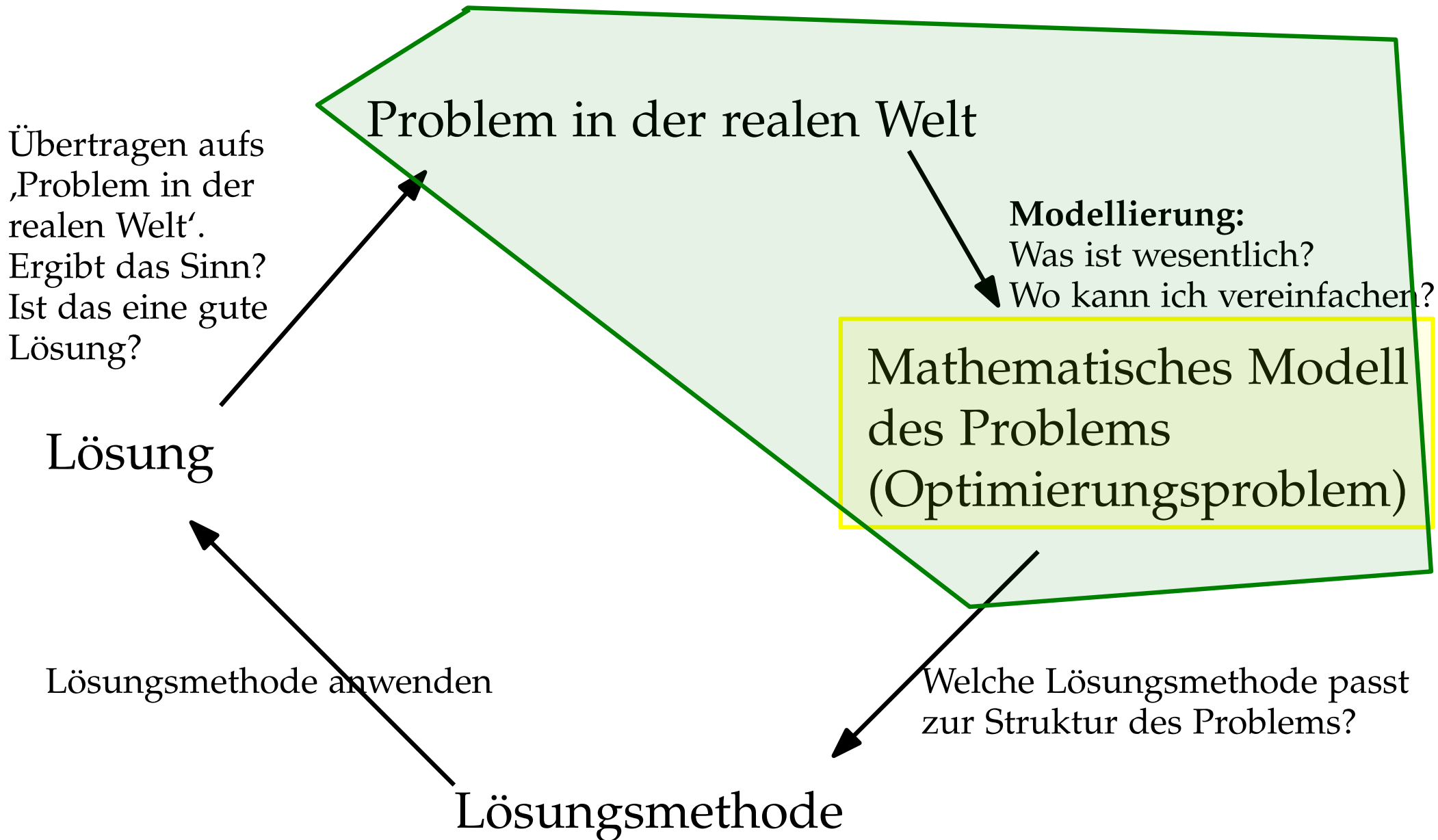
- Theodor-Boveri-Weg bis Am Hubland nehmen  
2 min (750 m)
- Am Galgenberg bis Rottendorfer Str. nehmen  
3 min (1,2 km)
- Rottendorfer Str. und Zweierweg bis Nürnberger Str. folgen  
4 min (1,5 km)
- Ohmstraße folgen  
2 min (1,0 km)

↪ Rechts abbiegen  
 Das Ziel befindet sich auf der linken Seite.  
12 Sek. (23 m)

**ROCK INN Boulderhalle**  
Ohmstraße 6, 97076 Würzburg



# Modellierungszyklus in der Optimierung



# Experimentelle Evaluation

Eingabe: Straßennetzwerk von Europa

- 18 Mio. Knoten, 42 Mio. Kanten



Algorithmus	Vorberechnung		Anfragen	
	Zeit [h:m]	Platz [GiB]	Zeit [ $\mu$ s]	Beschleun.
Dijkstra [Dij59]	—	—	2 550 000	—
ALT [GH05, GW05]	0:42	2.2	24 521	104
CRP [DGPW11]	$\ll$ 0:01	$<$ 0.1	1 650	1 545
Arc-Flags [Lau04]	0:20	0.3	408	6 250
CH [GSSD08]	0:05	0.2	110	23 181
TNR [ALS13]	0:20	2.1	1.25	2 040 000
HL [ADGW12]	0:37	18.4	0.56	4 553 571

# Experimentelle Evaluation

Eingabe: Straßennetzwerk von Europa

- 18 Mio. Knoten, 42 Mio. Kanten



Algorithmus	Vorberechnung		Anfragen	
	Zeit [h:m]	Platz [GiB]	Zeit [ $\mu$ s]	Beschleun.
Dijkstra [Dij59]	—	—	2 550 000	—
ALT [GH05, GW05]	0:42	2.2	24 521	104
CRP [DGPW11]	$\ll$ 0:01	$<$ 0.1	1 650	1 545
Arc-Flags [Lau04]	0:20	0.3	408	6 250
CH [GSSD08]	0:05	0.2	110	23 181
TNR [ALS13]	0:20	2.1	1.25	2 040 000
HL [ADGW12]	0:37	18.4	0.56	4 553 571

Danke an die Kollegen vom KIT für das Teilen dieser (und der folgenden) Folie!





Ittai Abraham, Daniel Delling, Andrew V. Goldberg, and Renato F. Werneck.

Hierarchical Hub Labelings for Shortest Paths.

In Leah Epstein and Paolo Ferragina, editors, Proceedings of the 20th Annual European Symposium on Algorithms (ESA'12), volume 7501 of Lecture Notes in Computer Science, pages 24–35. Springer, 2012.



Julian Arz, Dennis Luxen, and Peter Sanders.

Transit Node Routing Reconsidered.

In Proceedings of the 12th International Symposium on Experimental Algorithms (SEA'13), volume 7933 of Lecture Notes in Computer Science, pages 55–66. Springer, 2013.



Daniel Delling, Andrew V. Goldberg, Thomas Pajor, and Renato F. Werneck.

Customizable Route Planning.

In Panos M. Pardalos and Steffen Rebennack, editors, Proceedings of the 10th International Symposium on Experimental Algorithms (SEA'11), volume 6630 of Lecture Notes in Computer Science, pages 376–387. Springer, 2011.



Edsger W. Dijkstra.

A Note on Two Problems in Connexion with Graphs.

Numerische Mathematik, 1(1):269–271, 1959.

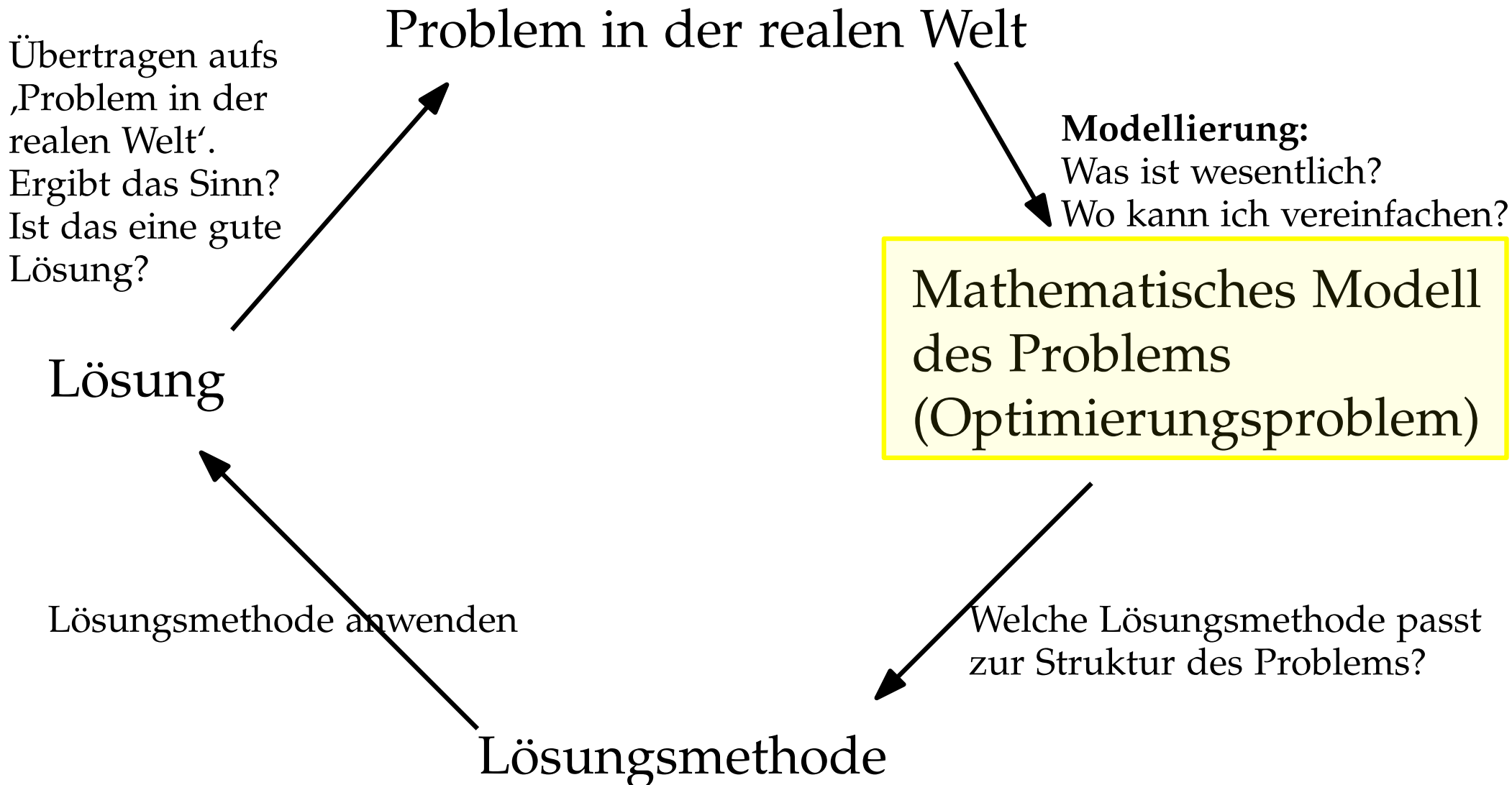


Andrew V. Goldberg and Chris Harrelson.

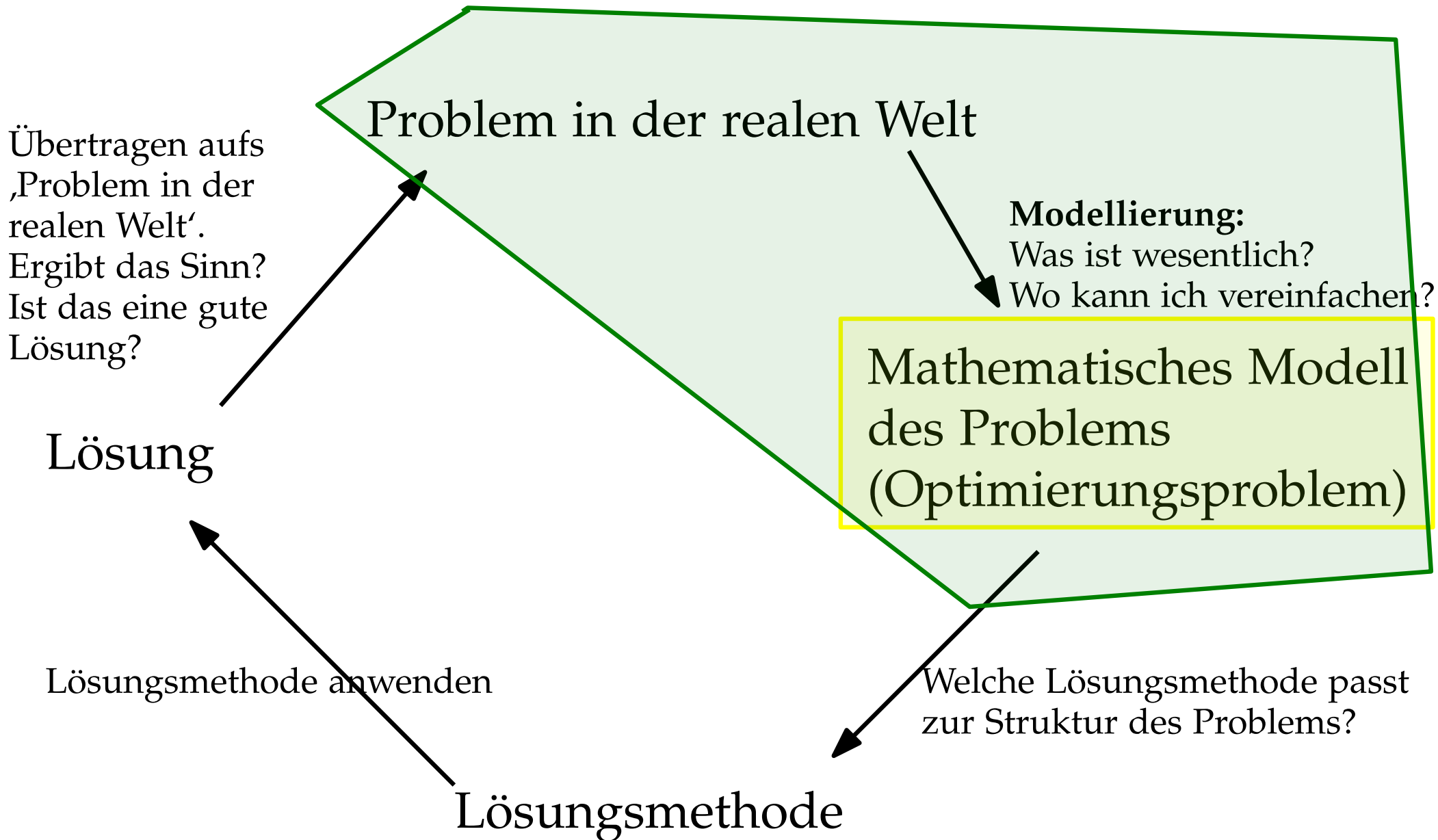
Computing the Shortest Path: A\* Search Meets Graph Theory.

In Proceedings of the 16th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA'05), pages 156–165. SIAM, 2005.

# Modellierungszyklus in der Optimierung



# Modellierungszyklus in der Optimierung



# Beispiel zur Modellierung in Graphen 2



# Beispiel zur Modellierung in Graphen 2

Wie kommen wir zu dieser Angebotsauswahl?



The screenshot displays two search results for rides from Würzburg to Berlin. Each result shows the departure time, duration, arrival time, price, driver profile, and a share icon.

Time	Origin	Duration	Destination	Price	Driver	Rating
14:40	Würzburg	5:40	Berlin	26,00 €	Brit Marie	5,0
15:10	Würzburg	4:50	Wustermark	23,00 €	Jurgen	

# Stichworte heute

Konzepte: Graph, Knoten, Kante, Kantenlabel, zusammenhängend, Subgraph, Weg, Matching, Clique

Optimierungsprobleme: Kürzeste Wege, kostenminimales Matching, größte Clique

Modellierung: Routenplanung, Anfragenmatching im Routesharing