Technische Universität Braunschweig

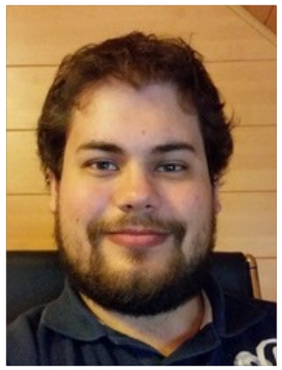
Institut für Betriebssysteme und Rechnerverbund
Algorithmik

Netzwerkalgorithmen

Übung 0: Organisation, Einleitung, Rückblick, Ausblick,...

Christian Rieck, 22. April 2021

Das Team



Vorlesung

Dr. Arne Schmidt // aschmidt@ibr.cs.tu-bs.de

Dienstags, 11:30 - 13:00, wöchentlich, Youtube & Discord

Große Übung

Christian Rieck // rieck@ibr.cs.tu-bs.de

Donnerstags, 11:30 - 13:00, ~ zweiwöchentlich, BBB

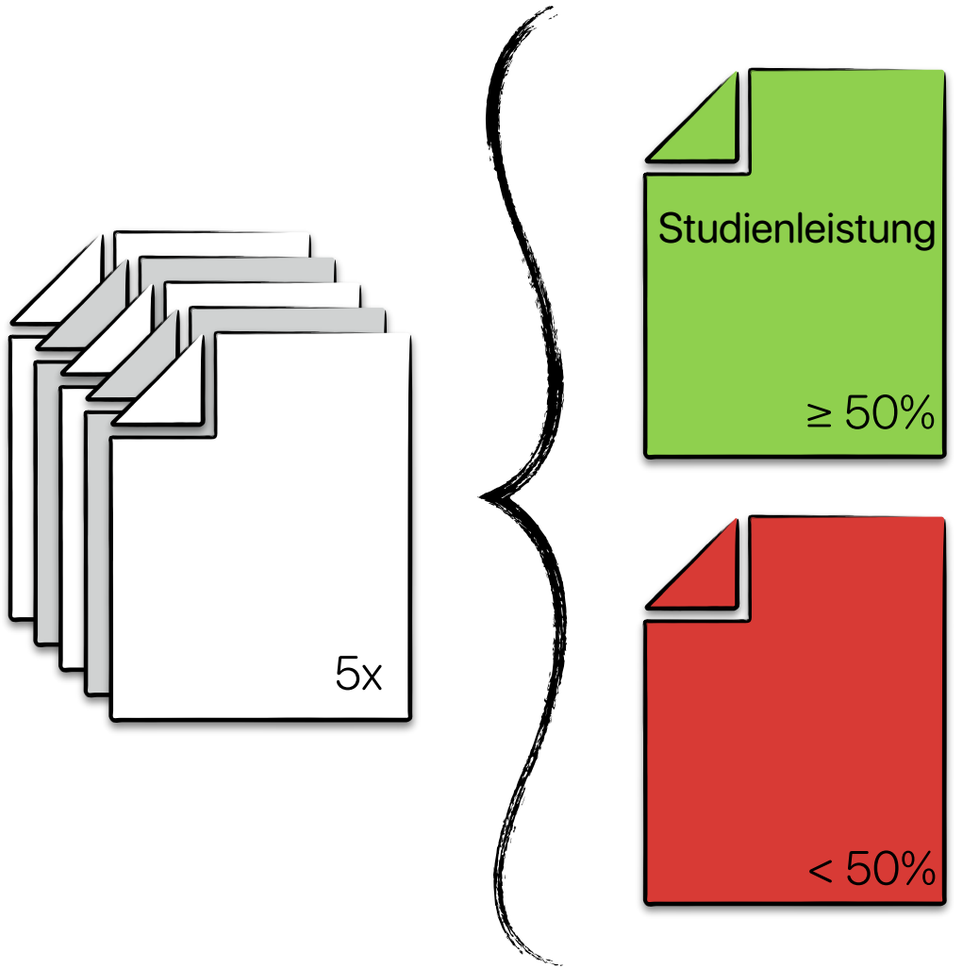
Kleine Übung

Marlin Melansek // melansek@ibr.cs.tu-bs.de

Tobias Wallner // wallner@ibr.cs.tu-bs.de

Freitags, 11:30 - 13:00, ~ zweiwöchentlich, Discord

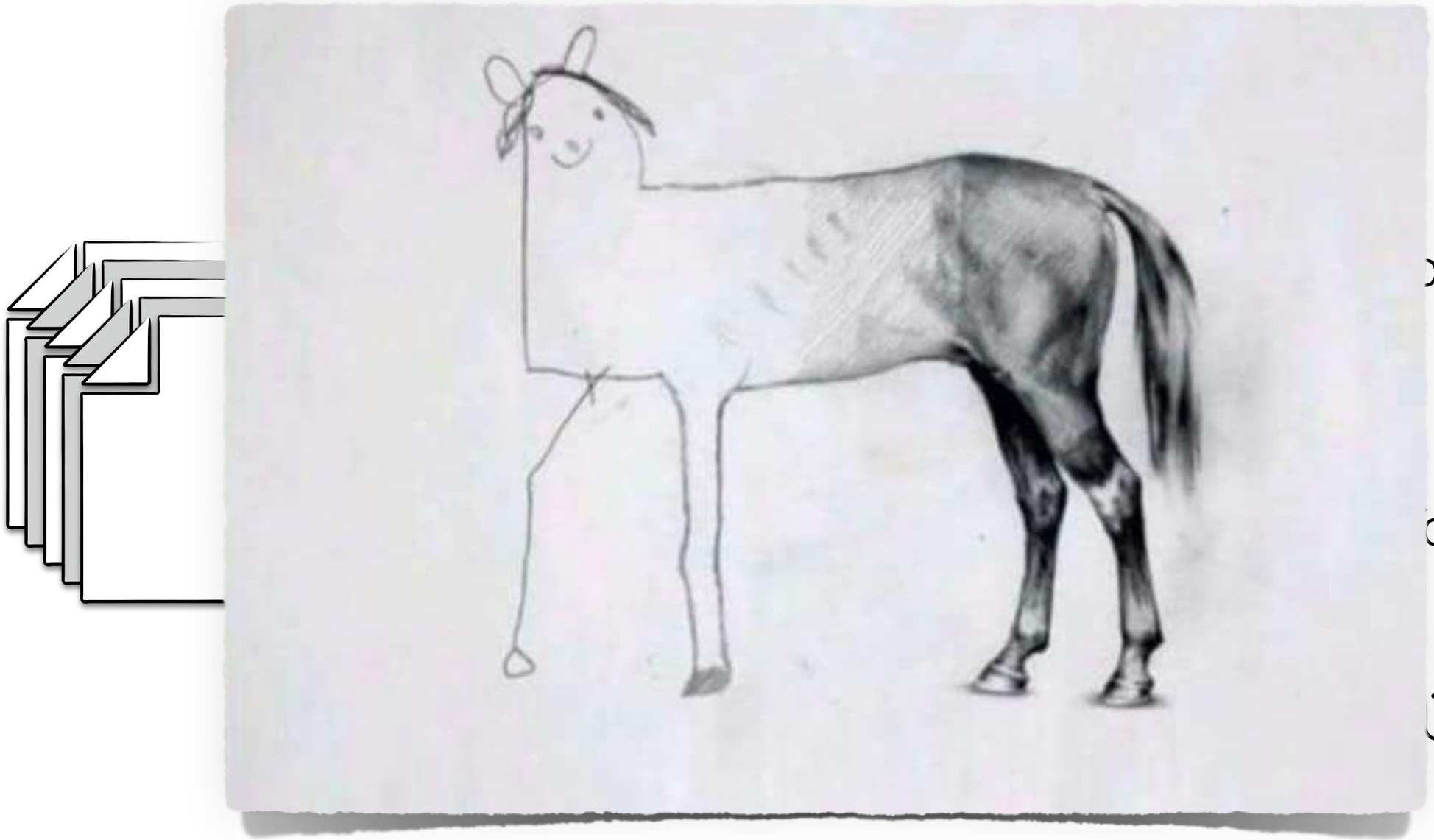
Hausaufgaben



- Aufgabenmix
  - Theorie / Beweise
  - Anwendung der Algorithmen
- alle zwei Wochen
- Ausgabe: Montags
- Abgabe: Sonntags
  - per Mail an Marlin / Tobias
- Rückgabe: Freitags
  - per Mail an euch
- Besprechung in kleiner Übung

https://www.ibr.cs.tu-bs.de/alg/Merkzettel/homework-booklet.pdf

Hausaufgaben



…orithmen

…obias

…Übung

https://www.ibr.cs.tu-bs.de/alg/Merkzettel/homework-booklet.pdf

Semesterplan

KW	Woche	Vorlesung (Di. 11:30)	Übung (Do. 11:30)	HA Ausgabe (Mo.)	HA Abgabe (So.)	HA Besprechung (Fr. 11:30)
15	12.04.21	-				
16	19.04.21	1	0	0		
17	26.04.21	2			0	
18	03.05.21	3	1	1		0
19	10.05.21	4			1	
20	17.05.21	5	2	2		1
21	24.05.21	Pfingstwoche				
22	31.05.21	6			2	
23	07.06.21	7		3		2
24	14.06.21	8	3		3	
25	21.06.21	9		4		3
26	28.06.21	10	4		4	
27	05.07.21	11		5		4
28	12.07.21	12	5		5	
29	19.07.21	13				5

Mailingliste: https://mail.ibr.cs.tu-bs.de/mailman/listinfo/na

Webseite: https://www.ibr.cs.tu-bs.de/courses/ss21/na/index.html

...zuletzt in *...

Graphen

Graph $G = (V, E)$

- Knotenmenge V
- Kantenmenge $E \subset \binom{V}{2}$

Graphen in NWA sind

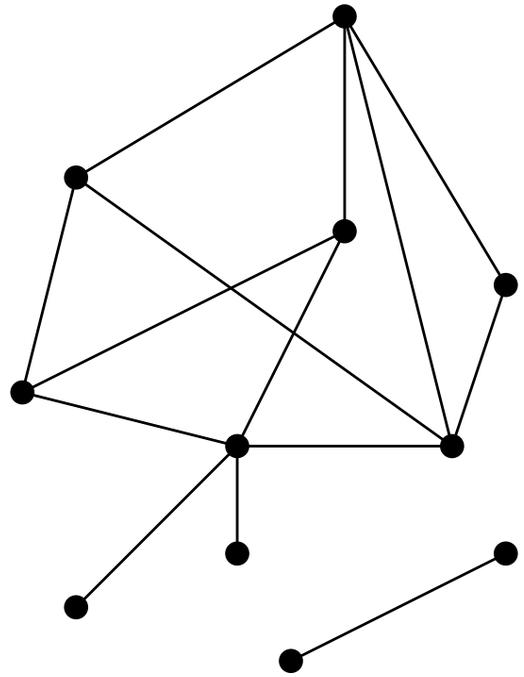
★ einfach:

Schleifen Mehrfachkanten

★ endlich: $|V| \in \mathbb{N}$

Zeichnung in der Ebene

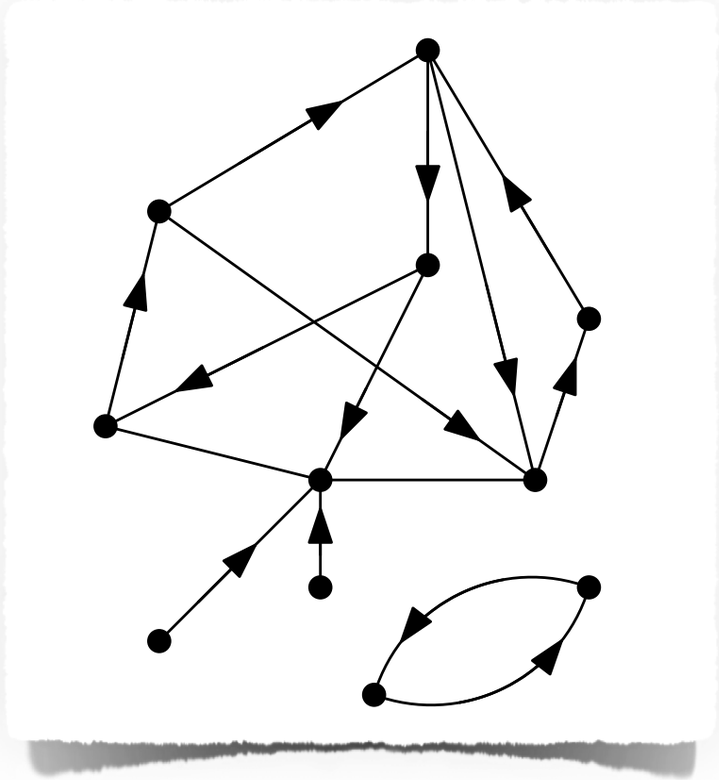
- Knoten → Punkt
- Kante → Kurve



... und gerichtete Graphen (Digraphen)

Gerichteter Graph $G = (V, E)$

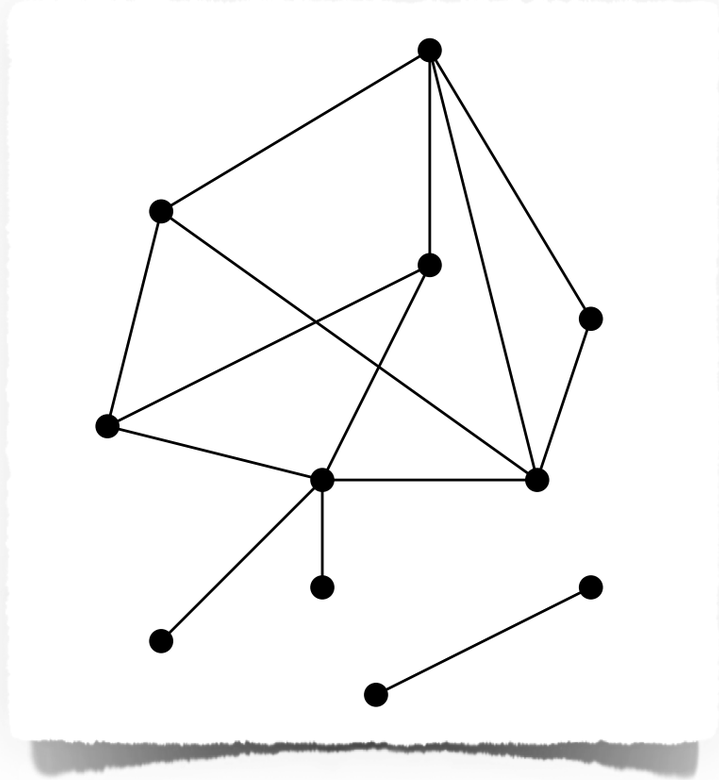
- Knotenmenge V
- Kantenmenge $E \subset V \times V$



$e = (u, v) \in E$

Graph $G = (V, E)$

- Knotenmenge V
- Kantenmenge $E \subset \binom{V}{2}$

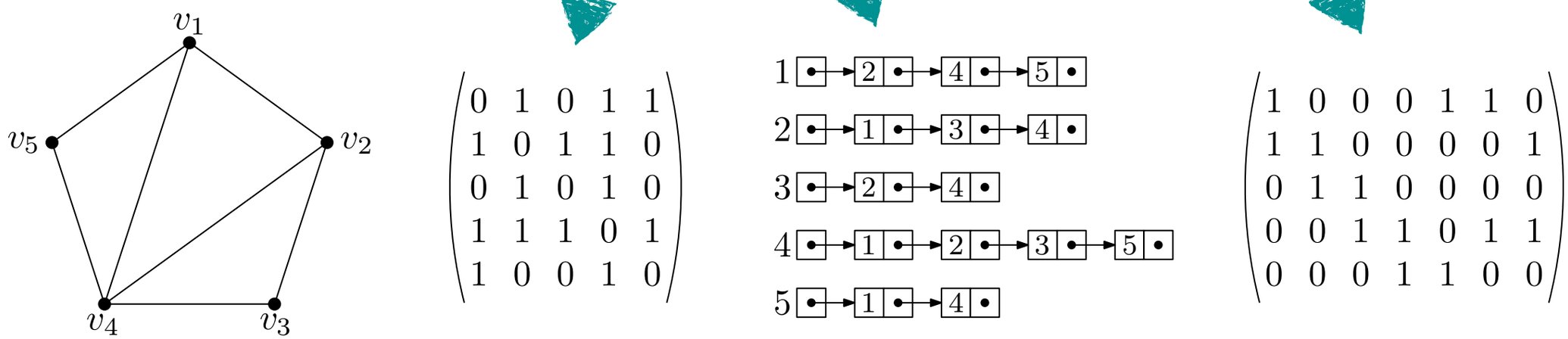


$e = \{u, v\} \in E$

Speicherung von Graphen

Es gibt verschiedene Möglichkeiten Graphen zu speichern/repräsentieren, z.B.

- Inzidenzmatrix
- Adjazenzliste
- Adjazenzmatrix

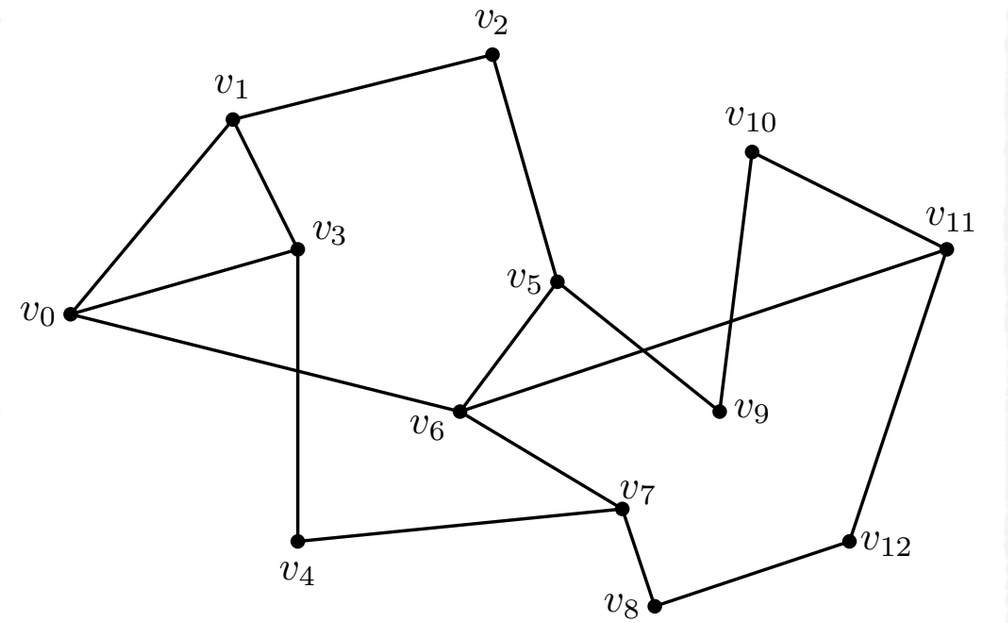


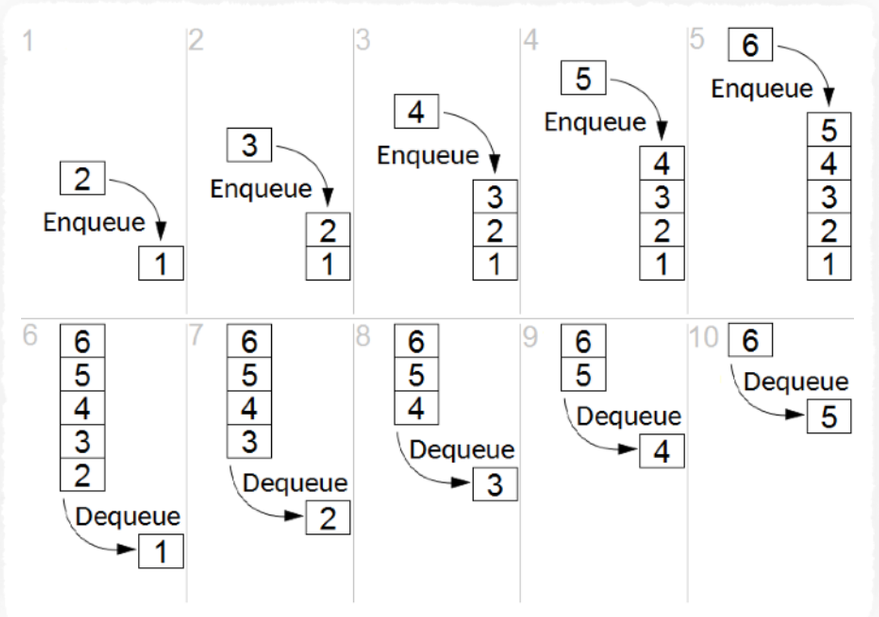
Macht euch damit vertraut; eine Anlaufstelle ist die Vorlesung AuD.

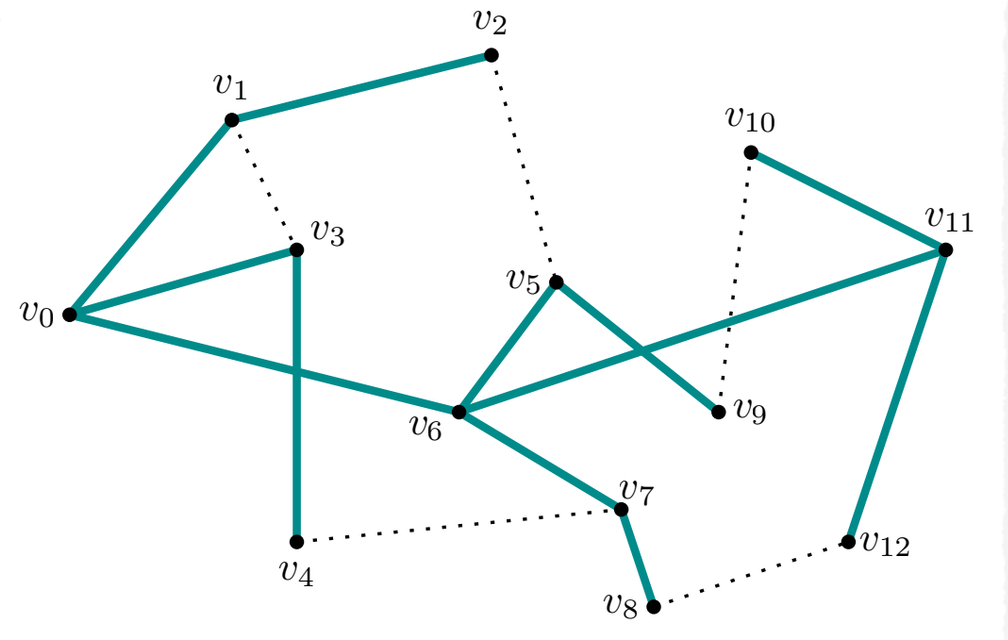
https://www.ibr.cs.tu-bs.de/courses/ws2021/aud/webpages/skript/VL8b.pdf
https://www.ibr.cs.tu-bs.de/courses/ws2021/aud/webpages/skript/VL9b.pdf

Breitensuche

```
procedure BFS(G,v)
    let Q be a queue
    Q.enqueue(v)
    while Q is not empty
        v = Q.dequeue()
        for each edge e incident to v
            let w be the other endpoint of e
            if w is not marked
                mark w
                Q.enqueue(w)
```

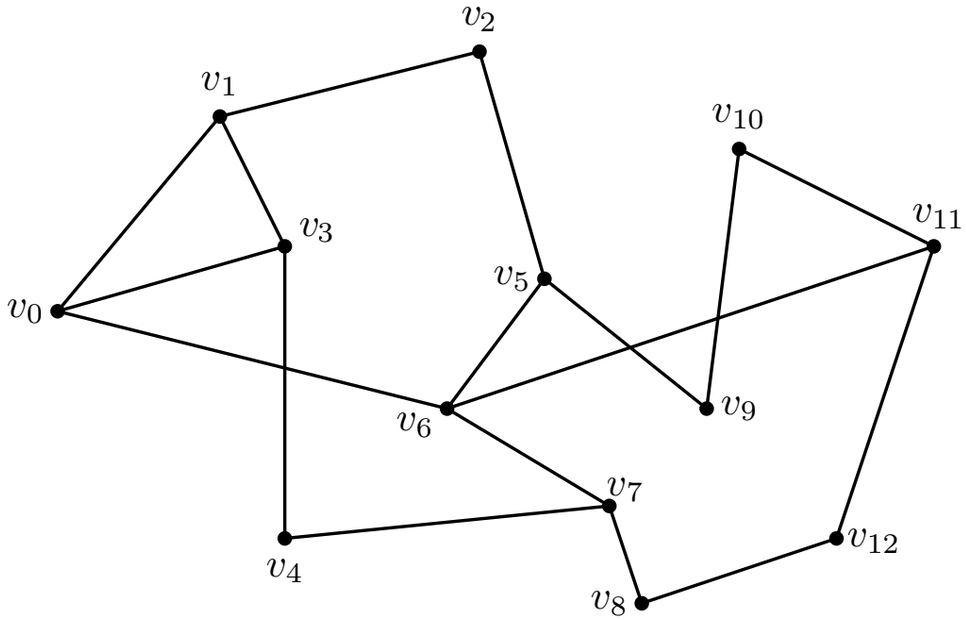


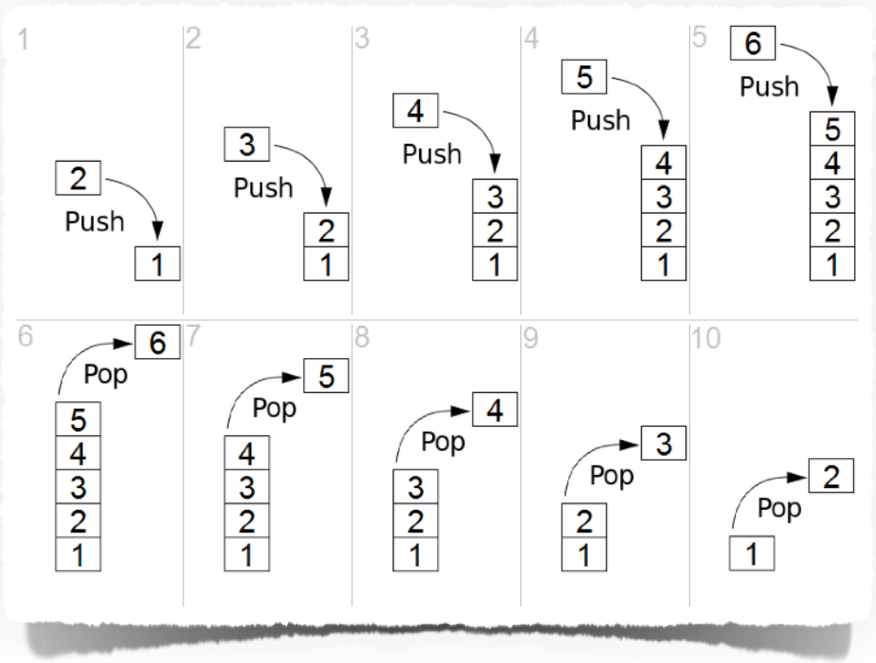


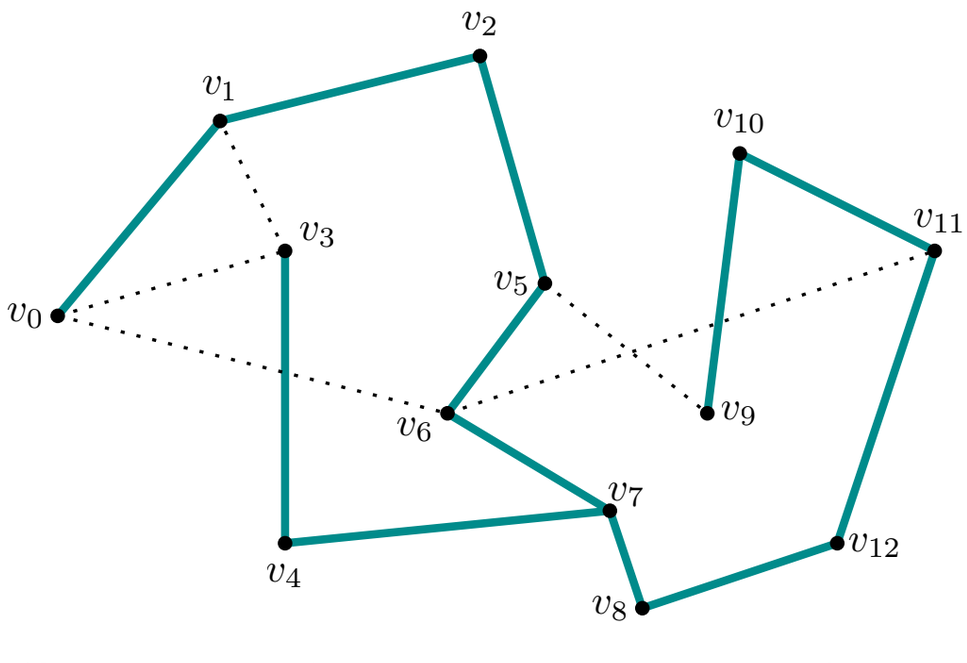


Tiefensuche

```
procedure DFS(G,v)
    let S be a stack
    S.push(v)
    while S is not empty
        v = S.pop()
        if v is not labeled as discovered
            label v as discovered
            for all edges from v to w
                S.push(w)
```







Asymptotisches Wachstum

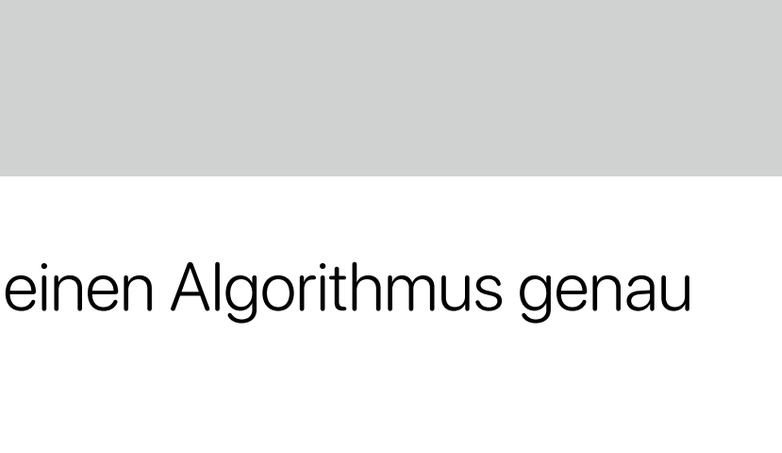
Da es oft schwierig bis unmöglich ist, den Zeitaufwand für einen Algorithmus genau anzugeben, schätzen wir den Aufwand asymptotisch ab.

O-Notation — obere Schranke: Die Menge an Funktionen, die asymptotisch höchstens so schnell wachsen wie f(n). Formal:

$$O(f(n)) := \{g : \mathbb{N} \to \mathbb{R} \mid \exists c \in \mathbb{R}^+, \exists n_0 \in \mathbb{N} : g(n) \leq c \cdot f(n),\ \forall n \geq n_0\}$$

Es gibt noch die Ω-Notation (untere Schranke) und die Θ-Notation (untere und obere Schranke).

Macht euch damit vertraut! Gute Anlaufstellen gibt es hier...



https://www.ibr.cs.tu-bs.de/alg/Merkzettel/runtime-booklet.pdf
https://www.ibr.cs.tu-bs.de/courses/ws2021/aud/webpages/skript/VL9b.pdf
https://www.ibr.cs.tu-bs.de/courses/ws2021/aud/webpages/uebungen/U4.pdf

Mathematische Aussagen und Beweise

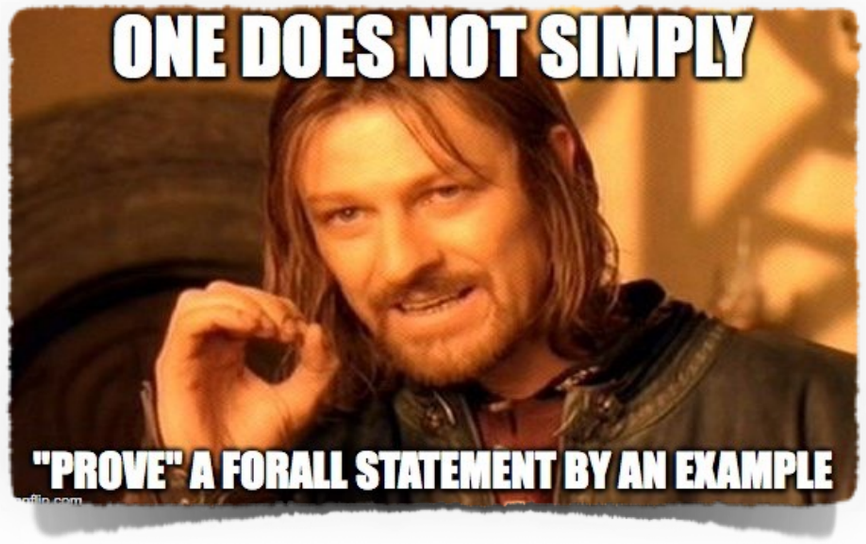
Eine (mathematische) Aussage ist ein Satz der entweder wahr oder falsch ist. Ein Beweis ist eine logisch vollständige Begründung dieser Aussagen.

Beweise von Aussagen sind essenziell. Es gibt dazu verschiedene Techniken, wie zum Beispiel Induktion, Kontraposition oder auch der direkte Beweis.

Oftmals ist es sehr anspruchsvoll etwas zu beweisen und es erfordert sehr viel Übung und Verständnis.

Macht euch mit diesen Techniken vertraut!
Gute Anlaufstellen gibt es hier...

https://www.ibr.cs.tu-bs.de/alg/Merkzettel/proof-booklet.pdf
https://www.ibr.cs.tu-bs.de/courses/ws1819/aud/uebungen/U2.pdf
https://www.ibr.cs.tu-bs.de/courses/ws1819/aud/uebungen/U3.pdf



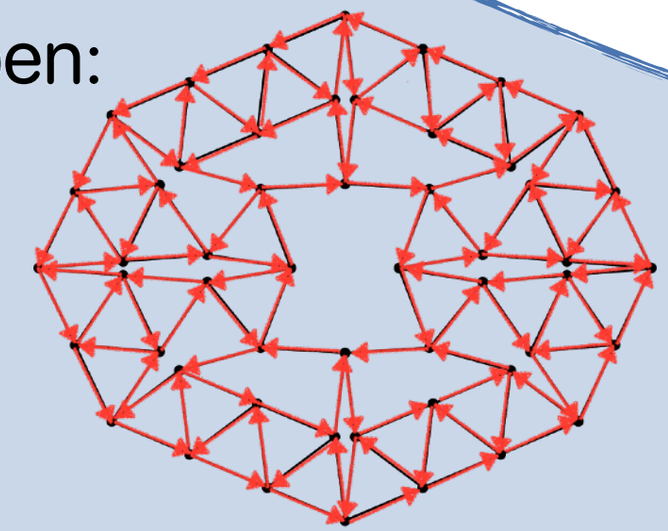
Problem vs. Instanz

Dies ist ein Problem!

Fleurys Algorithmus!

Gegeben: Ein Graph G
Gesucht: Eine Tour, die jede Kante genau einmal besucht.

Gegeben:



Dies ist eine Instanz!

Gesucht: Eine Tour, die jede Kante genau einmal besucht.

Konkrete Lösung!

Entscheidung vs. Optimierung

Hamiltonian Circle

Gegeben: Ein Graph

Gesucht: Ein Kreis der jeden Knoten genau einmal besucht

Entscheidungsproblem

Gegeben: Ein Graph

Gesucht: Eine spezielle Struktur oder ein Argument, dass eine solche Struktur in diesem Graphen nicht existiert

Traveling Salesman Tour

Gegeben: Ein gewichteter Graph

Gesucht: Ein Kreis minimalen Gewichts der jeden Knoten genau einmal besucht

Optimierungsproblem

Gegeben: Ein Graph mit Kantengewichten

Gesucht: Von allen speziellen Strukturen die zulässig sind, diejenige, die die Beste ist oder ein Argument, dass eine solche Struktur in diesem Graphen nicht existiert

Basiswissen ▷ Netzwerkalgorithmen

Version – 16. April 2021

Wissen welches wir für die Vorlesung voraussetzen:

- Graphen (Definition, spezielle Klassen, Teilgraphen, Speicherung,...)
- (Asymptotische) Laufzeiten von Algorithmen
- Grundlegende Graphenalgorithmen (BFS, DFS)
- Beweistechniken
- Sonstiges (Problem vs. Instanz, Optimierung vs. Entscheidung,...)

https://www.ibr.cs.tu-bs.de/alg/Merkzettel/nwa-booklet.pdf

Hausaufgabenblatt 0

Netzwerkalgorithmen **Sommersemester 2021**

Dr. Arne Schmidt
Christian Rieck

Übungsblatt 0

Dieses Blatt dient der eigenen Vorbereitung auf die Vorlesung und muss nicht abgegeben werden. Alle Aufgaben werden in der kleinen Übung am 07. Mai 2021 besprochen. Wir empfehlen euch allerdings, die Aufgaben vorher schon selbständig zu bearbeiten.

Ein Übungsblatt welches eigenverantwortlich grundlegende Kenntnisse testet:

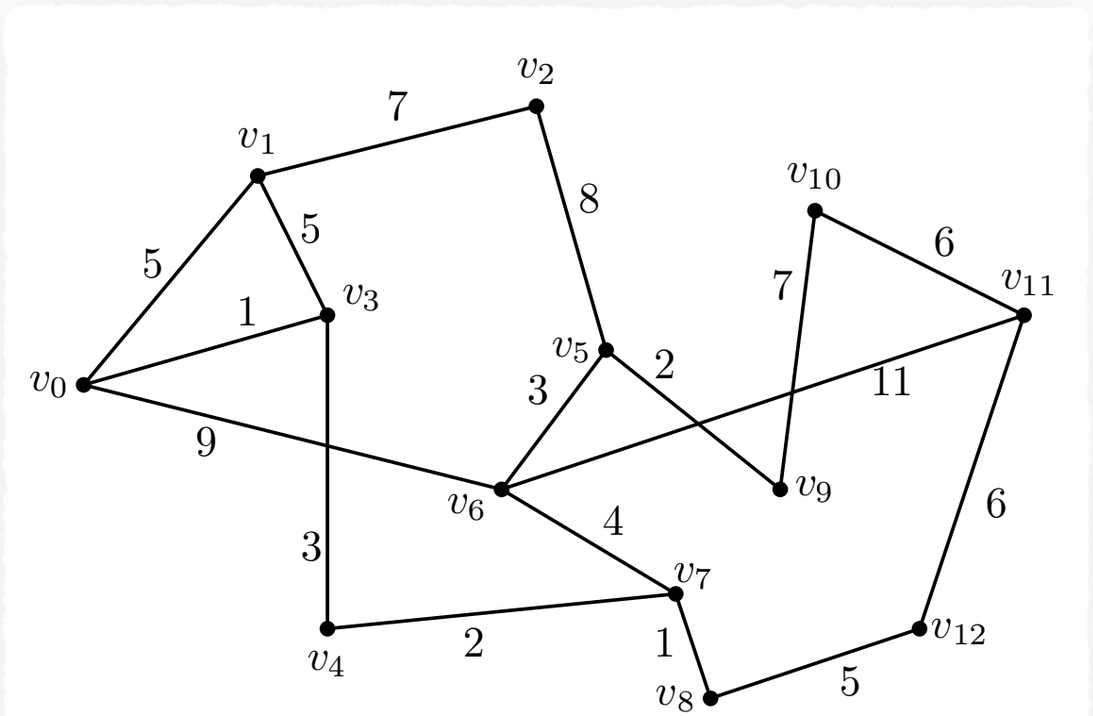
- Darstellung von Graphen
- Teilgraphen
- Grundlegende Eigenschaften von Graphen
- Graphenisomorphie

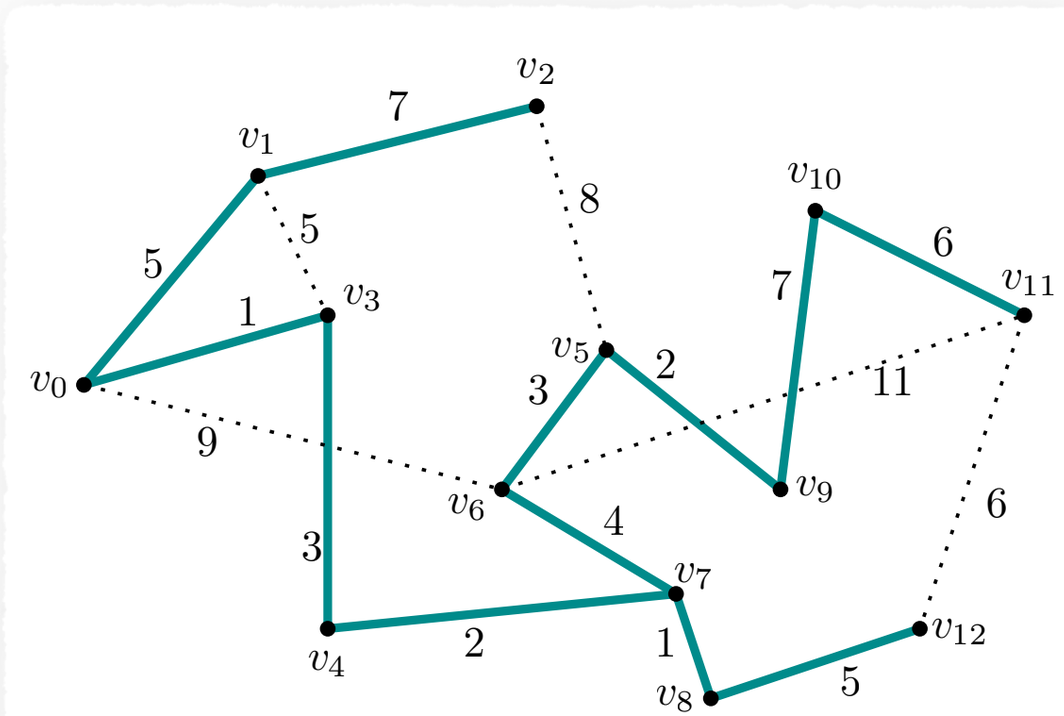
https://www.ibr.cs.tu-bs.de/courses/ss21/na/aufgaben/blatt0.pdf

...demnächst in Netzwerkalgorithmen...

Problem: Minimal aufspannende Bäume

Gegeben: Ein Graph mit Kantengewichten
Gesucht: Ein aufspannender Baum minimalen Gesamtgewichts

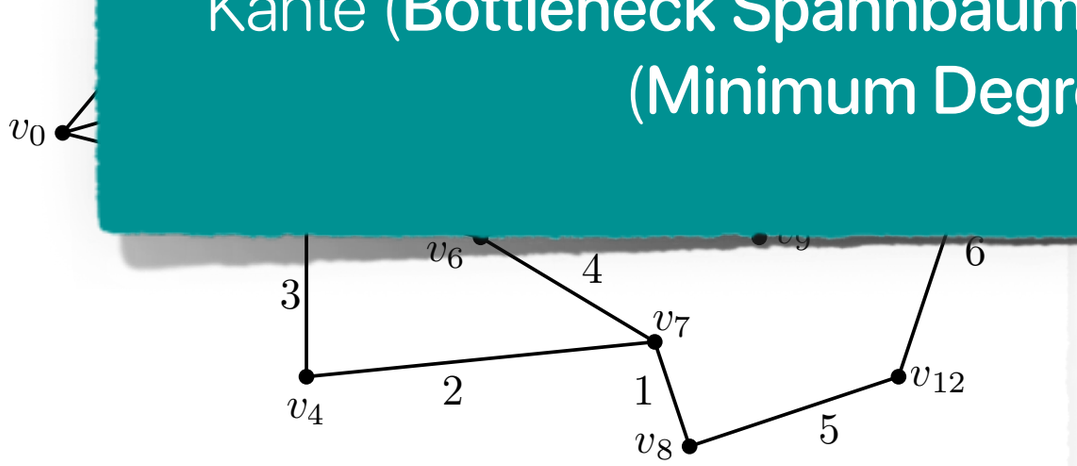


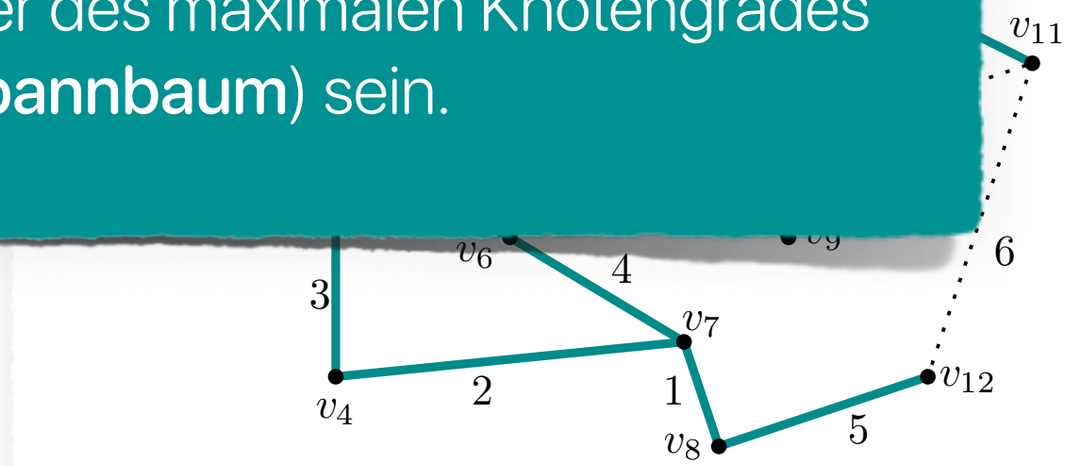


Problem: Minimal aufspannende Bäume

Die Verallgemeinerung ist das **Steinerbaum-Problem**. Geometrisch stellt sich die Frage nach einem aufspannenden Baum minimalen Gewichtes für eine gegebene Punktmenge (**Euklidischer MST**).

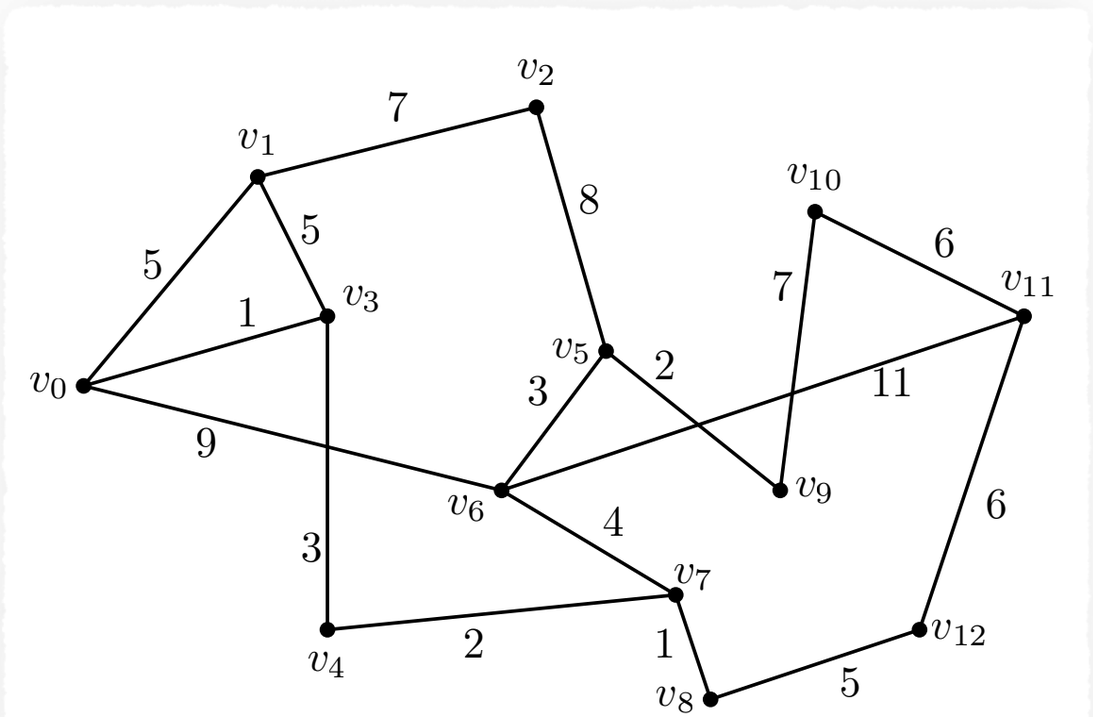
Weitere Bedingungen könnten zum Beispiel das Minimieren der längsten Kante (**Bottleneck Spannbaum**), oder des maximalen Knotengrades (**Minimum Degree Spannbaum**) sein.

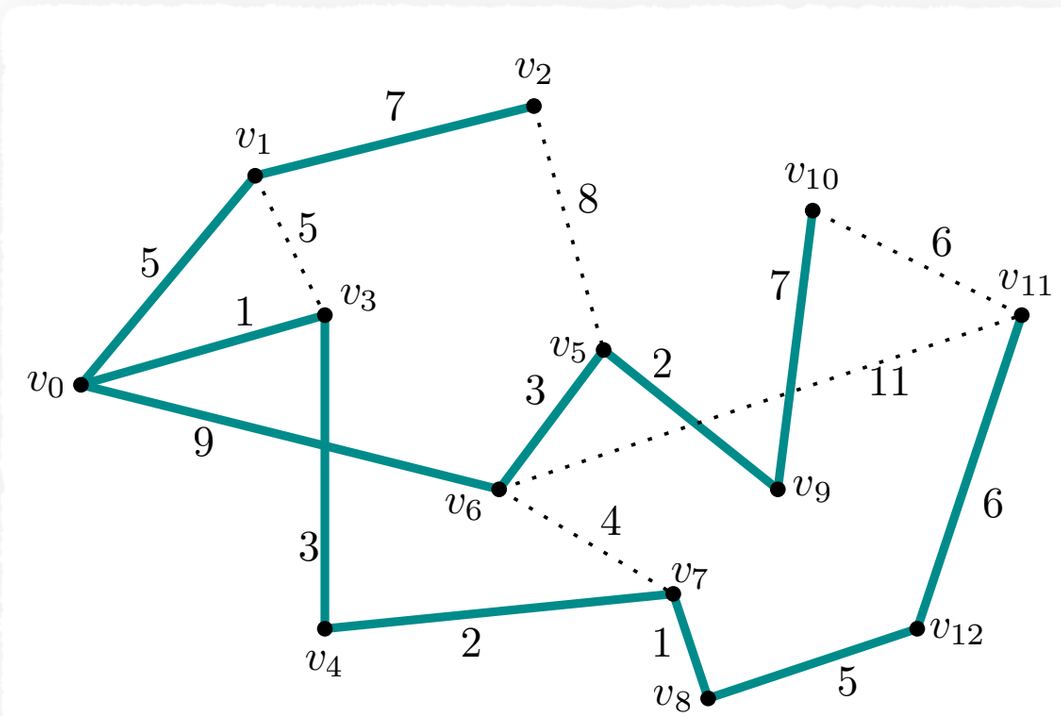




Problem: Kürzeste Wege

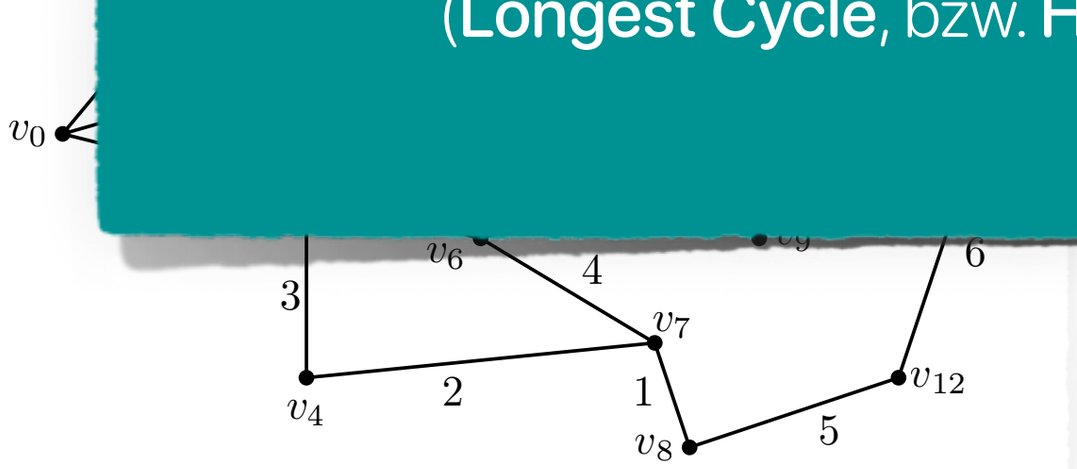
Gegeben: Ein (Di-)Graph mit Kantengewichten, ein Knoten v_0
Gesucht: Ein kürzester Wege Baum mit Wurzel v_0

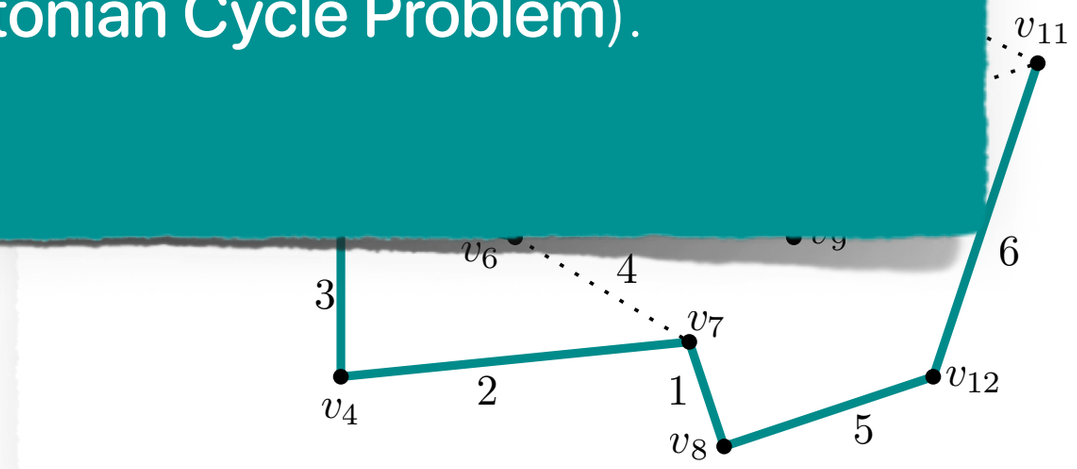




Problem: Kürzeste Wege

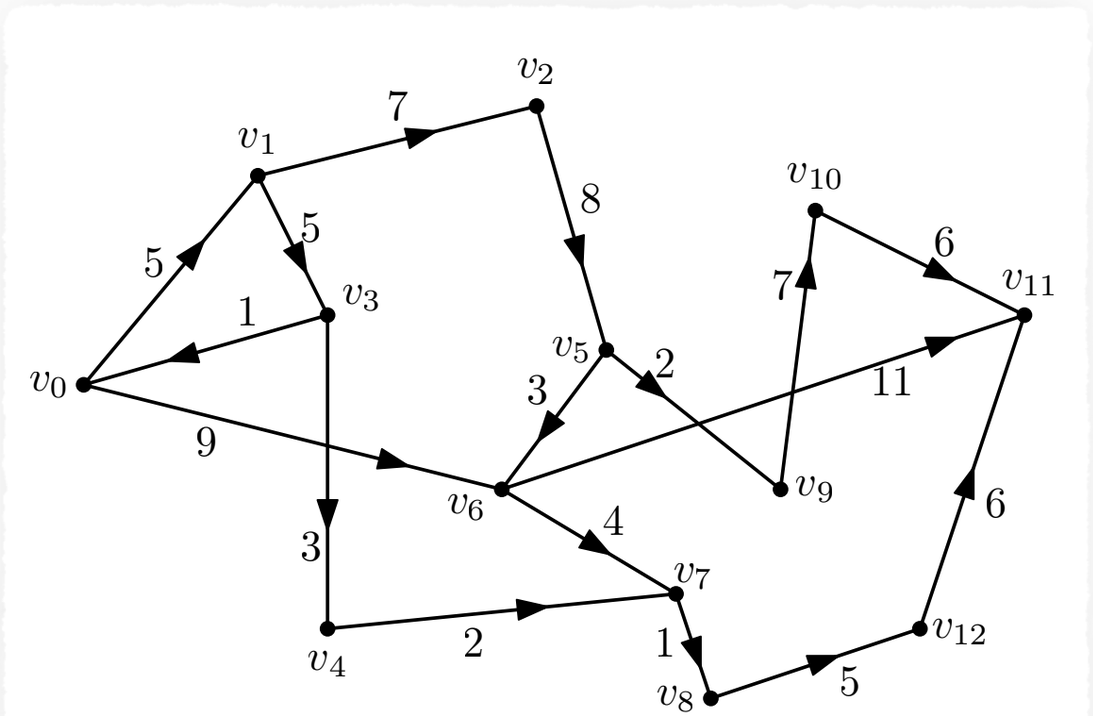
Weitere Probleme sind zum Beispiel die Frage nach längsten Pfaden (**Longest Path**, bzw. **Hamiltonian Path Problem**), die Frage nach kürzesten bzw. längsten Touren (**Traveling Salesman Problem**, bzw. **Maximum TSP**) oder einfach nach längsten Kreisen (**Longest Cycle**, bzw. **Hamiltonian Cycle Problem**).

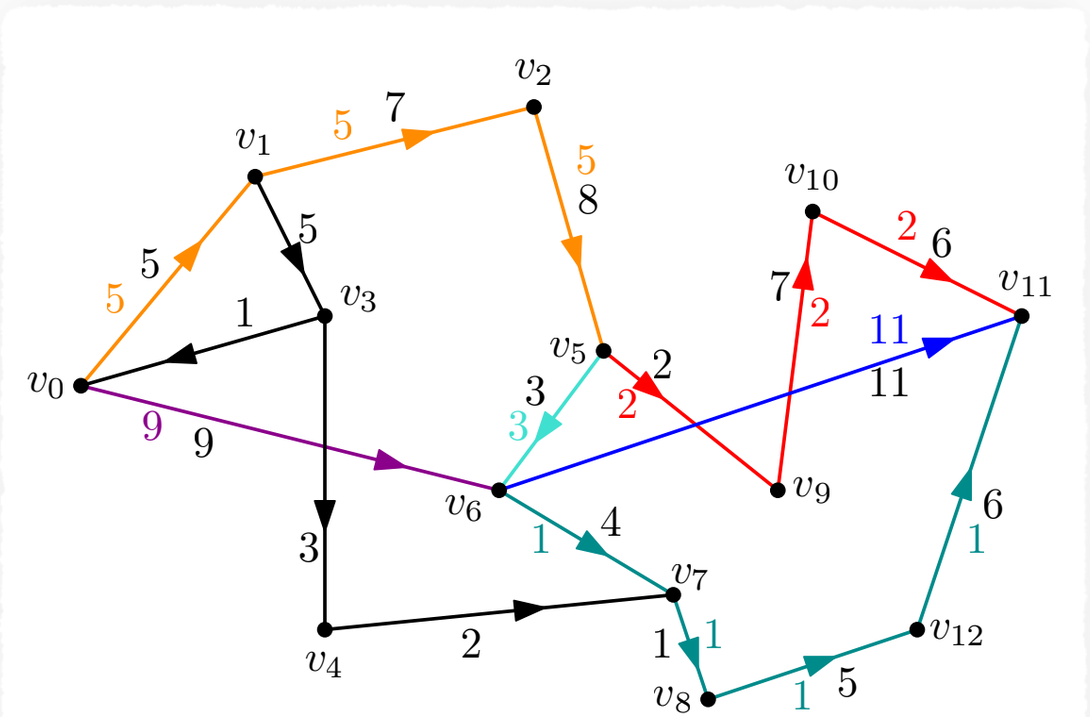




Problem: Maximaler Fluss

Gegeben: Ein Digraph mit Kantengewichten, Knoten s und t
Gesucht: Ein maximaler Fluss F von der Quelle s zur Senke t

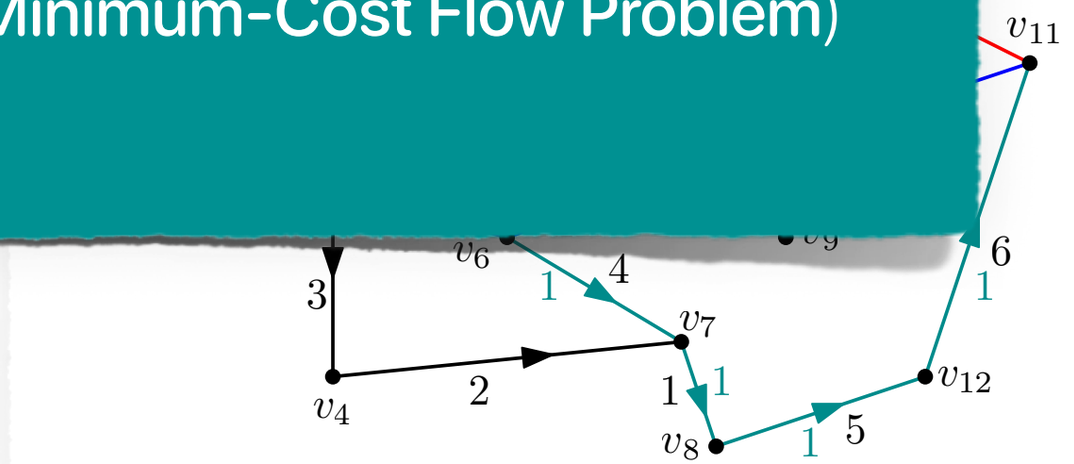




Problem: Maximaler Fluss

Verallgemeinerungen sind das **Zirkulationsproblem** oder die Berechnung von Flüssen mit mehreren Gütern (**Multi-Commodity Flow Problem**).

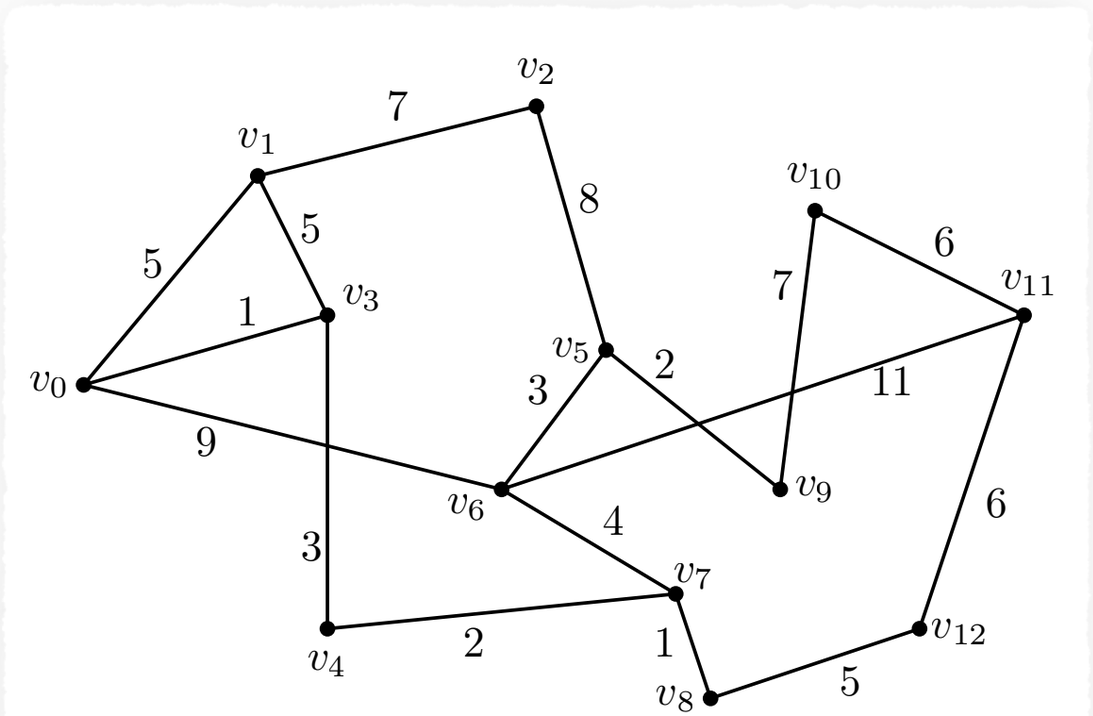
Es lassen sich Varianten betrachten die Kosten für das Verwenden von verschiedenen Kanten minimieren (**Minimum-Cost Flow Problem**)

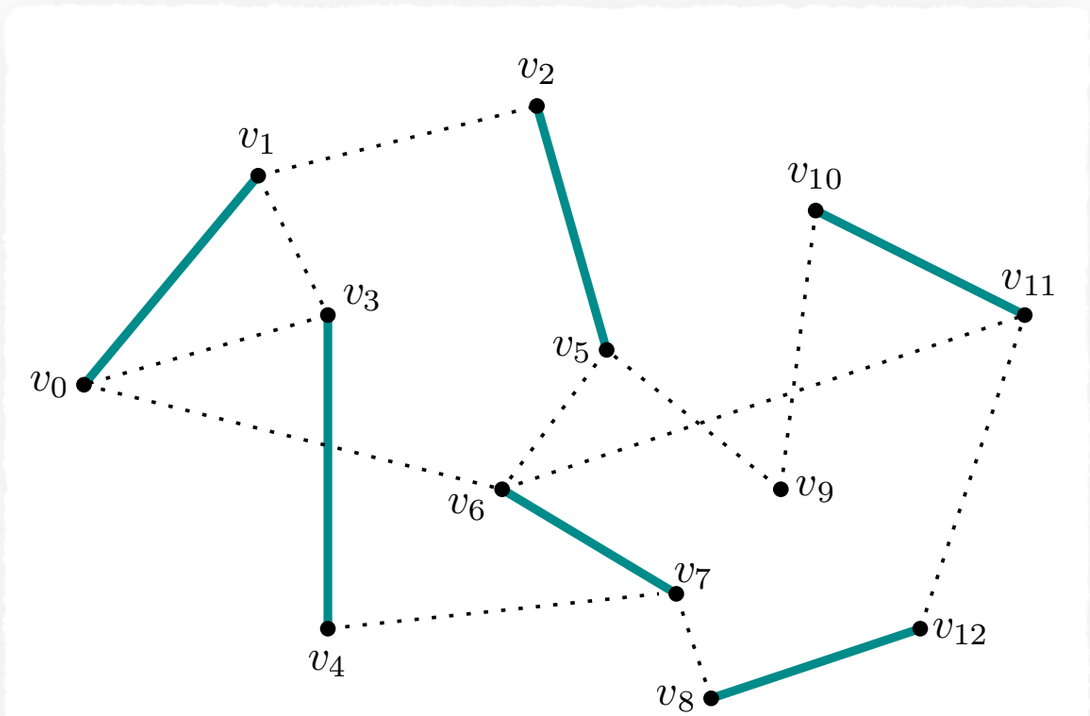


Problem: Maximales Matching

Gegeben: Ein Graph

Gesucht: Eine maximale Menge paarweise unabhängiger Kanten





Problem: Maximales Matching

Weitere Zielfunktionen sind zum Beispiel Matchings mit minimalen Kosten in gewichteten Graphen (**Minimum Cost Perfect Matching**) oder überdeckende Matchings in bipartiten Graphen (**Bipartite Matching**).

Es gibt weitere Verallgemeinerungen und Varianten wie **stabile Matchings** oder **3-dimensionale Matchings**.