Universität Basel

# *Algorithmische Mathematik*

*Graphen & Anwendungen*

**Skript zur Vorlesung**
**im**
**FS 2025**

Marc Schmidlin, Michael Multerer & Helmut Harbrecht

Version vom 5. März 2025

# Informationen ...

## zu den Übungen

Auf den Übungsblättern kommen Theorie- und Programmieraufgaben vor. Die Ausgabe und Abgabe der Übungsblätter findet jeweils am Montag statt. Für die Theorieaufgaben gelten folgende Regeln:

- Die Lösungen sind handschriftlich (Papier oder Tablet) zu schreiben.
- Die Abgaben erfolgen dabei entweder physisch an der Spiegelgasse 1 oder digital über ADAM durch jeweils **ein** PDF-File pro Übungsblattserie. Abgegebene PDF-Files müssen direkt A4-Papier druckbar sein.

Für die Programmieraufgaben gelten folgende Regeln:

- Die Lösungen sind in `MATLAB` zu programmieren.
- Die Abgabe der Codes erfolgt dabei immer digital über ADAM durch jeweils **ein** ZIP-File pro Übungsblattserie. Die ZIP-Files enthalten dabei jeweils alle notwendigen Sourcecode-Files. Falls in den Programmieraufgaben Fragen zu beantworten sind, werden allfällige Antworten mit den Theorieaufgaben abgegeben.

## zu der Leistungsüberprüfung

Die Vorlesung *Algorithmische Mathematik: Graphen und Anwendungen* wird als eine kombinierte Veranstaltung "Vorlesung mit Übung" (6KP) mit lehrveranstaltungs-begleitender Leistungsüberprüfung gehalten, welche wie folgt angedacht ist:

- In den Übungsblätter sind 50% der Punkte, von den abzugebenden Übungsaufgaben, zu erreichen, um die Zulassung für die mündliche Prüfung zu erhalten.
- Es ist *keine* schriftliche Prüfung vorgesehen.
- Die mündliche Prüfung (Dauer 30min) wird benotet (Skala 1-6 0,5). Die mündlichen Prüfungen finden vom 04-06.06.2025 statt.

Wenn Sie die Zulassung für die mündliche Prüfung erhalten, wird die Veranstaltung für Sie mit der Note der mündlichen Prüfung benotet. Haben Sie jedoch die Zulassung für die mündliche Prüfung nicht erhalten, dürfen Sie nicht an der mündlichen Prüfung teilnehmen und die Veranstaltung wird für Sie mit einer ungenügenden Note bewertet.

# Vorwort

Diese Mitschrift kann und soll nicht ganz den Wortlaut der Vorlesung wiedergeben. Sie soll das Nacharbeiten des Inhalts der Vorlesung erleichtern.

*Literatur zur Vorlesung:*

- H. Harbrecht und M. Multerer: *Algorithmische Mathematik: Graphen, Numerik und Probabilistik*, Springer Spektrum
- N. Blum: *Algorithmen und Datenstrukturen*, Oldenbourg-Verlag
- B. Korte und J. Vygen: *Combinatorial Optimization: Theory and Algorithms*, Springer-Verlag

# Inhaltsverzeichnis

# 1 GRUNDLAGEN

## 1.1 Graphen

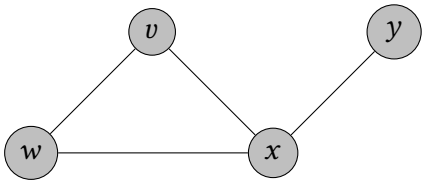
**Definition 1.1** Ein *Graph* ist ein Paar $G = (V, E)$, bestehend aus endlicher Menge $V$ von *Knoten* (vertices) und einer endlichen Menge von ungeordneten Paaren $\{v, w\} \subseteq V$,

$$E \subseteq \big\{ X \subseteq V \,:\, |X| = 2 \big\},$$

den *Kanten* (edges). Eine Kante $e = \{v, w\} \in E$ steht für eine Verbindungen zwischen den verschiedenen Knoten $v$ und $w$, da $\{v, w\} = \{w, v\}$ gilt, ist dieser Verbindung keine Richtung zugeordnet; sprich die Kante $e$ ist ungerichtet.

*Achtung:* Wie zumindest in einigen Bereichen in der mathematischen Literatur nicht unüblich, kann die Terminologie in der Graphentheorie variieren. Dabei kann der gleiche Term, je nach Quelle, verschiedene (aber eben auch ähnliche) Bedeutungen annehmen und, umgekehrt verschiedene Terme exakt die gleiche Bedeutung haben. In der Literatur wird zum Beispiel, was hier als Graph bezeichnt wird, auch *ungerichteter Graph* genannt.

**Beispiel 1.2** Graph $G = \big(\{v, w, x, y\}, \big\{\{v, w\}, \{w, x\}, \{v, x\}, \{x, y\}\big\}\big)$:




♣

**Definition 1.3** Sei $G = (V, E)$ ein Graph. Ein *Weg* in $G$ ist eine Knotenfolge

$$\pi = v_0, v_1, \ldots, v_r$$

mit $r \geq 1$ und $\{v_i, v_{i+1}\} \in E$, $i = 0, 1, \ldots, r-1$. Wir sprechen von einem Weg, der *von $v$ nach $w$* führt, wenn der Anfangsknoten $v_0 = v$ und der Endknoten $v_r = w$ ist. Die *Länge* $|\pi|$ von $\pi$ ist $r$, das heisst, die Anzahl der Kanten, die in $\pi$ durchlaufen werden. Ein Knoten $w$ heisst von einem Knoten $v$ *erreichbar*, falls ein Weg $\pi = v_0, v_1, \ldots, v_r$ in $G$ existiert, so dass $v = v_0$ und $w = v_r$.

*Beachte:* In der Literatur wird ein Weg auch *Pfad* genannt.

**Beispiel 1.4** Beispiele für Wege in dem Graphen von Beispiel 1.2 sind

- $\pi_1 = v, w$ (Länge 1),
- $\pi_2 = v, w, x$ (Länge 2),
- $\pi_3 = v, w, x, y, x, y$ (Länge 5) und
- $\pi_4 = x, v, w, x, y$ (Länge 4). ♣

**Definition 1.5** Sei $G = (V, E)$ ein Graph und ein Weg

$$\pi = v_0, v_1, \dots, v_r$$

mit $r \geq 1$ gegeben, dann nennen wir den Weg $\pi$ einen *Rundweg*, falls $v_0 = v_r$ gilt.

**Definition 1.6** Sei $G = (V, E)$ ein Graph und ein Weg

$$\pi = v_0, v_1, \dots, v_r$$

mit $r \geq 1$ gegeben. Der Weg $\pi$ heisst *einfach*, falls gilt:

- $v_0, v_1, \dots, v_r$ sind paarweise verschieden
- oder $\pi$ ist ein Rundweg und $v_0, v_1, \dots, v_{r-1}$ sind paarweise verschieden.

**Beispiel 1.7** Der Graph aus Beispiel 1.2 besitzt die einfachen Rundwege $\pi_1 = v, w, x, v$ und $\pi_2 = x, y, x$. Dementgegen ist $\pi_3 = v, w, x, y, x, v$ ein Rundweg, welcher nicht einfach ist. ♣

**Definition 1.8** Es sei $G = (V, E)$ ein Graph und $v \in V$. Wir definieren:

- die Menge der *(direkten) Nachbarn* von $v$

$$\text{post}(v) := \{w \in V : \{v, w\} \in E\},$$

- die Menge aller von $v$ erreichbaren Knoten

$$\text{post}^\star(v) := \{w \in V : \text{es existiert ein Weg von } v \text{ nach } w\}.$$

**Beispiel 1.9** Für den Graphen aus Beispiel 1.2 ist

$$\text{post}(x) = \{v, w, y\}, \qquad \text{post}^\star(x) = \{v, w, x, y\}.$$ ♣

**Definition 1.10** Es sei $G = (V, E)$ ein Graph und $v \in V$. Der *Grad* von $v$ ist gegeben durch

$$\deg(v) := |\text{post}(v)|.$$

*Beachte:* Insbesondere gilt

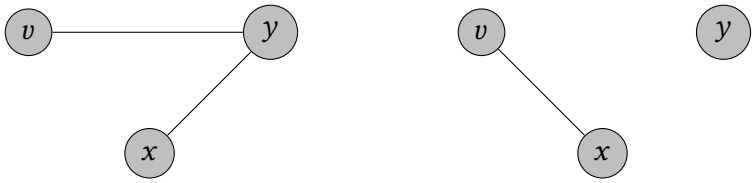
$$\sum_{v \in V} \deg(v) = 2|E|.$$

**Definition 1.11** Es sei $G = (V, E)$ ein Graph. Ein Graph $G' = (V', E')$ heisst *Teilgraph von* $G$, falls $V' \subseteq V$ und $E' \subseteq E$ gelten.

*Beachte:* Da $G'$ selber ein Graph sein muss, gilt

$$E' \subseteq E \cap \{X \subseteq V' \,:\, |X| = 2\}.$$

**Beispiel 1.12** Der Graph rechts ist ein Teilgraph des Graphs von Beispiel 1.2, der links keiner:



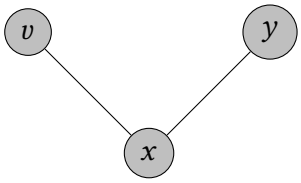

♣

**Definition 1.13** Es sei $G = (V, E)$ ein Graph. Für eine Knotenteilmenge $V' \subseteq V$ definieren wir den *von $V'$ induzierten Teilgraphen von $G$*, durch $G[V'] = (V', E')$ mit

$$E' := E \cap \left\{ X \subseteq V' \,:\, |X| = 2 \right\}.$$

*Beachte:* Offensichtlich ist jeder induzierter Teilgraph immer ein Teilgraph.

**Beispiel 1.14** Der von $\{v, x, y\}$ induzierte Teilgraph des Graphs von Beispiel 1.2 ist:




♣

**Definition 1.15** Es seien $G = (V, E)$ und $G' = (V', E')$ zwei Graphen. Die Graphen $G$ und $G'$ heissen *isomorph*, falls es eine Abbildung $\phi : V \to V'$ gibt, wobei

- $\phi$ bijektiv ist und
- $E' = \left\{ \{\phi(v), \phi(w)\} \,:\, \{v, w\} \in E \right\}$ erfüllt ist.

Eine solche Abbildung $\phi$ nennen wir einen *Graphenisomorphismus* von $G$ und $G'$.

Grundlegend am Graphenisomorphismus ist, dass er die Graphenstruktur des einen Graphen exakt auf den anderen abbildet. Das heisst wenn man nur die Graphenstruktur betrachtet kann man zwei isomorphe Graphen nicht unterscheiden: in dem Sinne sie sind also lediglich unterschiedliche Beschreibungen der gleichen abstrakten Graphenstrukur.

**Beispiel 1.16** Der Graph $G = \big(\{v, w, x, y\}, \big\{\{v, w\}, \{w, x\}, \{v, x\}, \{x, y\}\big\}\big)$ ist isomorph zu dem Graphen $G' = \big(\{b, d, a, c\}, \big\{\{a, b\}, \{b, d\}, \{a, c\}, \{a, d\}\big\}\big)$. ♣

Um zu überprüfen, ob zwei Graphen isomorph sind, muss man einen Isomorphismus zwischen den Graphen finden. A priori kommt gemäss Definition jede bijektive Abbildung zwischen $V$ und $V'$ potenziell in Frage. Folgende Aussage schränkt die potenziellen bijektiven Abbildung $\phi : V \to V'$ für einen allfälligen Graphenisomorphismus weiter ein.

**Satz 1.17** *Seien $G = (V, E)$ und $G' = (V', E')$ zwei isomorphe Graphen. Dann erfüllt jeder Graphenisomorphismus $\phi : V \to V'$*

$$\deg(v) = \deg\big(\phi(v)\big)$$

*für alle $v \in V$.*

*Beweis.* Trivial. ♠

## 1.2 Digraphen

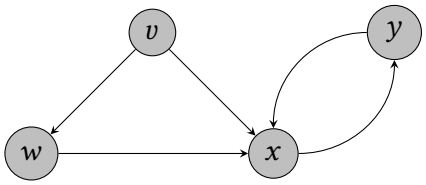
**Definition 1.18** Ein *Digraph* ist ein Paar $G = (V, E)$, bestehend aus einer endlicher Menge $V$ von *Knoten* (vertices) und einer endlichen Menge von geordneten Paaren $(v, w) \in V \times V$,

$$E \subseteq \big\{(v, w) \in V \times V \,:\, v \neq w\big\},$$

den *Kanten* (edges). Eine Kante $e = (v, w) \in E$ steht für eine Verbindungen zwischen den verschiedenen Knoten $v$ und $w$, da $(v, w) \neq (w, v)$ gilt, ist dieser Verbindung eine Richtung zugeordnet; sprich sie ist gerichtet. Wir nennen $v$ den *Anfangsknoten* und $w$ den *Endknoten* der Kante $e$.

*Beachte:* In der Literatur wird ein Digraph auch *gerichteter Graph* genannt.

**Beispiel 1.19** Digraph $G = \big(\{v, w, x, y\}, \{(v, w), (w, x), (v, x), (x, y), (y, x)\}\big)$:




♣

**Bemerkung** In Digraphen werden manchmal auch Kanten der Form $(v, v)$ zugelassen. Man spricht dann von *Schleifen*. ♦

**Definition 1.20** Sei $G = (V, E)$ ein Digraph. Ein *Weg* in $G$ ist eine Knotenfolge

$$\pi = v_0, v_1, \dots, v_r$$

mit $r \geq 1$ und $(v_i, v_{i+1}) \in E$, $i = 0, 1, \dots, r-1$. Wir sprechen von einem Weg, der *von $v$ nach $w$* führt, wenn der Anfangsknoten $v_0 = v$ und der Endknoten $v_r = w$ ist. Die *Länge* $|\pi|$ von $\pi$ ist $r$, das heisst, die Anzahl der Kanten, die in $\pi$ durchlaufen werden. Ein Knoten $w$ heisst von einem Knoten $v$ *erreichbar*, falls ein Weg $\pi = v_0, v_1, \dots, v_r$ in $G$ existiert, so dass $v = v_0$ und $w = v_r$.

**Beispiel 1.21** Beispiele für Wege in dem Digraphen von Beispiel 1.19 sind

- $\pi_1 = v, w$ (Länge 1),
- $\pi_2 = v, w, x$ (Länge 2) und
- $\pi_3 = v, w, x, y, x, y$ (Länge 5). ♣

**Definition 1.22** Sei $G = (V, E)$ ein Digraph und ein Weg

$$\pi = v_0, v_1, \dots, v_r$$

mit $r \geq 1$ gegeben, dann nennen wir den Weg $\pi$ einen *Rundweg*, falls $v_0 = v_r$ gilt.

**Definition 1.23** Sei $G = (V, E)$ ein Digraph und ein Weg

$$\pi = v_0, v_1, \dots, v_r$$

mit $r \geq 1$ gegeben. Der Weg $\pi$ heisst *einfach*, falls gilt:

- $v_0, v_1, \dots, v_r$ sind paarweise verschieden
- oder $\pi$ ist ein Rundweg und $v_0, v_1, \dots, v_{r-1}$ sind paarweise verschieden.

**Beispiel 1.24** Der Digraph aus Beispiel 1.19 besitzt als einzigen einfachen Rundweg den Rundweg $\pi_1 = x, y, x$ respektive mit $y$ als Anfangsknoten $\pi_1 = y, x, y$. In dem wir den Rundweg $\pi_1$ mehrmals aneinander hängen können wir die weiteren Rundwege $\pi_2 = y, x, y, x, y$, $\pi_3 = y, x, y, x, y, x, y$ etc. generieren. Diese sind aber jeweils per definition nicht einfach. Weiter gibt es keinen Rundweg, der die Knoten $v$ oder $w$ enthält. ♣

**Definition 1.25** Es sei $G = (V, E)$ ein Digraph und $v \in V$. Wir definieren:

- die Menge der *(direkten) Nachfolger* von $v$

$$\text{post}(v) := \{w \in V \,:\, (v, w) \in E\},$$

- die Menge der *(direkten) Vorgänger* von $v$

$$\mathrm{pre}(v) := \{w \in V \,:\, (w,v) \in E\},$$

- die Menge aller von $v$ erreichbaren Knoten

$$\mathrm{post}^\star(v) := \{w \in V \,:\, \text{es existiert ein Weg von } v \text{ nach } w\},$$

- die Menge aller Knoten, die $v$ erreichen können,

$$\mathrm{pre}^\star(v) := \{w \in V \,:\, v \in \mathrm{post}^\star(w)\}.$$

Ein Knoten $w \in \mathrm{post}(v) \cup \mathrm{pre}(v)$ ist ein *Nachbar* von $v$.

*Beachte:* $\mathrm{post}(v)$ und $\mathrm{pre}(v)$ können völlig verschieden sein, ebenso $\mathrm{post}^\star(v)$ und $\mathrm{pre}^\star(v)$.

**Beispiel 1.26** Für den Digraphen aus Beispiel 1.19 ist

$$\mathrm{post}(v) = \{w, x\}, \qquad \mathrm{post}^\star(v) = \{w, x, y\}, \qquad \mathrm{pre}(v) = \mathrm{pre}^\star(v) = \emptyset$$

und

$$\mathrm{post}(y) = \{x\}, \qquad \mathrm{post}^\star(y) = \{x, y\}.$$ ♣

**Definition 1.27** Es sei $G = (V, E)$ ein Digraph und $v \in V$. Der *Eingangsgrad* von $v$ ist gegeben durch

$$\mathrm{indeg}(v) := |\mathrm{pre}(v)|$$

und der *Ausgangsgrad* durch

$$\mathrm{outdeg}(v) := |\mathrm{post}(v)|.$$

Schliesslich definieren wir den *Grad* von $v$ vermittels

$$\deg(v) := \mathrm{indeg}(v) + \mathrm{outdeg}(v).$$

*Beachte:* Es gilt zwingend:

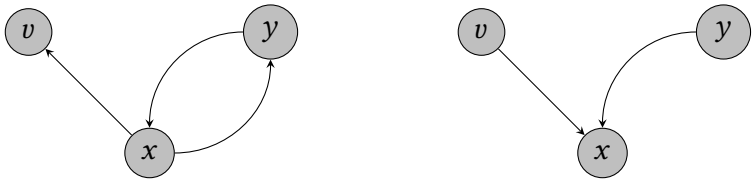
$$\sum_{v \in V} \mathrm{outdeg}(v) = \sum_{v \in V} \mathrm{indeg}(v) = |E|.$$

**Definition 1.28** Es sei $G = (V, E)$ ein Digraph. Ein Digraph $G' = (V', E')$ heisst *Teildigraph von* $G$, falls $V' \subseteq V$ und $E' \subseteq E$ gelten.

*Beachte:* Da $G'$ selber ein Digraph sein muss, gilt

$$E' \subseteq E \cap \big\{(v,w) \in V' \times V' \,:\, v \neq w\big\}.$$

**Beispiel 1.29** Der Digraph rechts ist ein Teildigraph des Digraphs von Beispiel 1.19, der links keiner:



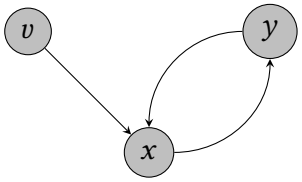

♣

**Definition 1.30** Es sei $G = (V, E)$ ein Digraph. Für eine Knotenteilmenge $V' \subseteq V$ definieren wir den *von $V'$ induzierten Teildigraphen von $G$*, durch $G[V'] = (V', E')$ mit

$$E' := E \cap \big\{(v,w) \in V' \times V' \,:\, v \neq w\big\}.$$

*Beachte:* Offensichtlich ist jeder induzierter Teildigraph immer ein Teildigraph.

**Beispiel 1.31** Der von $\{v, x, y\}$ induzierte Teildigraph des Digraphs von Beispiel 1.19 ist:




♣

**Definition 1.32** Es seien $G = (V, E)$ und $G' = (V', E')$ zwei Digraphen. Die Digraphen $G$ und $G'$ heissen *isomorph*, falls es eine Abbildung $\phi : V \to V'$ gibt, wobei

- $\phi$ bijektiv ist und
- $E' = \big\{(\phi(v), \phi(w)) \,:\, (v,w) \in E\big\}$ erfüllt ist.

Eine solche Abbildung $\phi$ nennen wir einen *Digraphenisomorphismus* von $G$ und $G'$.

**Satz 1.33** *Seien $G = (V, E)$ und $G' = (V', E')$ zwei isomorphe Digraphen. Dann erfüllt jeder Digraphenisomorphismus $\phi : V \to V'$*

$$\mathrm{indeg}(v) = \mathrm{indeg}\big(\phi(v)\big), \qquad \mathrm{outdeg}(v) = \mathrm{outdeg}\big(\phi(v)\big)$$

*für alle $v \in V$.*

*Beweis.* Trivial. ♠

**Definition 1.34** Es sei $G = (V, E)$ ein Digraph. Dann ist der von $G$ induzierte Graph $G'$ gegeben durch $G' = (V, E')$ mit

$$E' := \big\{\{v, w\} : (v, w) \in E\}.$$

**Beispiel 1.35** Der von dem Digraph von Beispiel 1.19 induzierte Graph ist genau der Graph von Beispiel 1.2. ♣

**Definition 1.36** Es sei $G = (V, E)$ ein Graph. Dann ist der von $G$ induzierte Digraph $G'$ gegeben durch $G' = (V, E')$ mit

$$E' := \big\{(v, w) : \{v, w\} \in E\}.$$

**Beispiel 1.37** Der von dem Graph von Beispiel 1.2 induzierte Digraph ist nicht der Digraph von Beispiel 1.19. Der Digraph von Beispiel 1.19 ist lediglich ein echter Teildigraph. ♣

## 1.3 Implementation von Graphen & Digraphen

Die einfachste Art Digraphen im Rechner zu speichern ist die Verwendung von *Adjazenzmatrizen*.

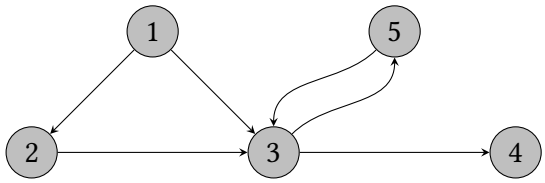
**Definition 1.38** Ein Digraph $G = (V, E)$ mit $V = \{1, 2, \dots, n\}$ kann durch eine *Adjazenzmatrix* oder *Nachbarschaftsmatrix* $\mathbf{A} = [a_{i,j}]_{i,j=1}^{n} \in \mathbb{R}^{n \times n}$ mit

$$a_{i,j} = \begin{cases} 1, & \text{falls } (i, j) \in E, \\ 0, & \text{sonst,} \end{cases}$$

dargestellt werden.

**Definition 1.39** Ein Graph $G = (V, E)$ mit $V = \{1, 2, \dots, n\}$ kann durch die Adjazenzmatrix des durch $G$ induzierten Digraphen dargestellt werden.

**Beispiel 1.40** Der Digraph $G$



und sein induzierten Graphen $G'$ besitzen die Adjazenzmatrizen

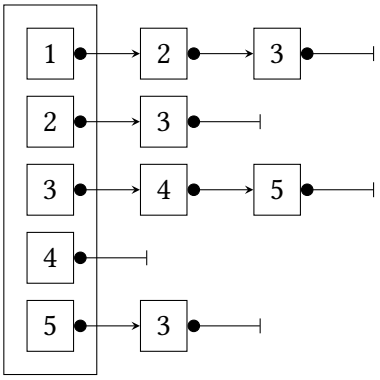
$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \qquad \mathbf{A}' = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \qquad \clubsuit$$

**Bemerkung** Bei Graphen ist $\mathbf{A}$ stets symmetrisch, das heisst es gilt $a_{i,j} = a_{j,i}$ für alle $1 \leq i, j \leq n$. ♦

Die Adjazenzmatrix besitzt immer den Speicherplatzbedarf $|V|^2$, unabhängig von der Anzahl $|E|$ der Kanten. Platzeffizienter ist die Darstellung von Digraphen $G = (V, E)$ durch *Adjazenzlisten*:

**Definition 1.41** Die *Adjazenzliste* oder *Nachbarschaftsliste* zu einem Knoten $v \in V$ enthält einen Knoten $w \in V$ genau dann, wenn $w \in \text{post}(v)$ gilt.

**Beispiel 1.42** Die Adjazenzlisten zum Digraphen aus Beispiel 1.40 werden wie folgt dargestellt:




♣

Adjazenzlisten können gespeichert werden als *einfach verkettete Listen* mit Elementen der Form

```
struct node
  {  unsigned int  number;
     struct node  *next;
  };
```

Sind `A` und `B` vom Typ `struct node`, so enthält `A.number` beziehungsweise `B.number` die Nummer des Knotens, während die Zuweisung

```
A.next = &B;
```

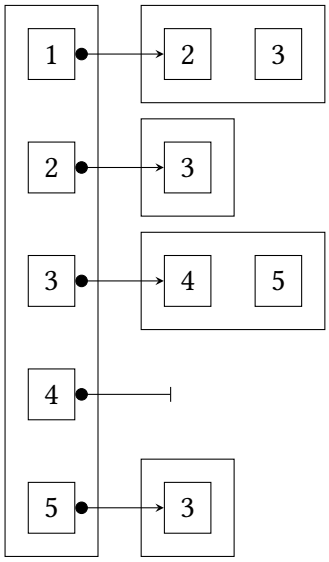
`B` als Nachfolger von `A` definiert. Anstelle von `B` kann man dann auch `A.next` schreiben. Das Ende der Liste wird durch

```
B.next = NULL;
```

markiert. Merken muss man sich jeweils den Anfang der Liste, auch *Listenkopf* (head) genannt. Dies geschieht meist durch ein Feld der Länge $|V|$.

Für Digraphen haben Adjazenzlisten den Seicherplatzbedarf $|V| + |E|$. Für Graphen ist der Platzbedarf $|V| + 2|E|$, da jede Kante in zwei Adjazenzlisten vorkommt.

**Bemerkung** Eine weitere Möglichkeit ist es, jede Adjazenzliste statt als einfach verkette Liste direkt als Feld zu speichern. Für einen Knoten $v \in V$ muss das Feld dann jeweils die Länge $\text{outdeg}(v)$ haben. Beispielsweise wuerden die Adjazenzlisten zum Digraphen aus Beispiel 1.40 dann wie folgt dargestellt werden:




In MATLAB kann man diese Variante umsetzen in dem man die Adjazenzlisten als Vektoren in ein `cell array` der Länge $|V|$ speichert. ♦

Unter der Annahme, dass der maximale Ausgangsgrad eines Digraphen nicht zu gross ist, kann man anstatt den einfach verketteten Listen oder den Vektoren, die Adjazenzliste auch als Spalten in eine rechteckige Matrix speichern. Dabei hat die Matrix dann $|V|$ Spalten und $\max_{v \in V} \text{outdeg}(v)$ Reihen. Um die Adjazenzlisten alle gleich lang zu machen ergänzt man sie mit Nullen. Für den Digraphen aus Beispiel 1.40 erhält man dann:

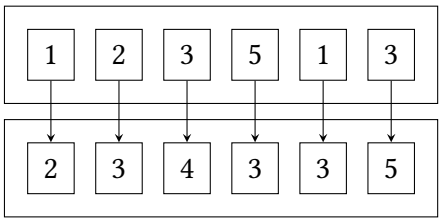
$$\begin{bmatrix} 2 & 3 & 4 & 0 & 3 \\ 3 & 0 & 5 & 0 & 0 \end{bmatrix}.$$

Während wir diese einfache Variante in den Uebungen benutzen werden, ist es auch angebracht zu bemerken, dass es auch raffiniertere Darstellungen von Digraphen gibt, welche insbesondere auch für die Speicherung von dünnvernetzen Digraphen gebräulich sind:

**Bemerkung**

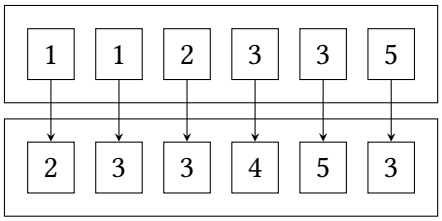
- Beim *coordinate list* (COO) Format speichert man in zwei Felder der Länge $|E|$ jeweils den Anfangsknoten in dem einen Feld und den Endknoten in dem

anderen Feld an der gleichen Stelle und dies für alle Kanten. Für den Digraphen aus Beispiel 1.40 hat man zum Beispiel:

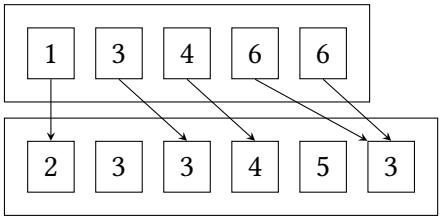



Ein Nachteil dieser Darstellung ist, dass man jeweils Suchen muss, um herauszufinden, ob eine Kante im Graph existiert oder nicht.

- Um das COO Format zu verbessern kann man die Reihenfolge sortieren. Dabei möchte man die Reihenfolge so wählen, dass die Anfangsknoten aufsteigend sortiert sind, und die Endknoten zum gleichen Anfangsknoten auch aufsteigend sortiert sind. Für den Digraphen aus Beispiel 1.40 hat man zum Beispiel:




- Im sortierten COO Format sieht man, dass die Sortierung der Anfangsknoten eine Redundanz offenbart. Es ist nicht nötig die Anfangsknoten als Feld zu merken, sondern es reicht sich für jeden Knoten zu merken, wo der potenziell erste Eintrag in dem Endknotenfeld ist. Dies ergibt genau das *compressed sparse row* (CSR) Format. Für den Digraphen aus Beispiel 1.40 hat man zum Beispiel:




Der Vorteil dieses Formats ist, dass es sehr speichereffizient ist und die Existenz von Kanten auch effizient bestimmt werden kann. Ein Nachteil ist, dass das Einfügen oder Löschen von Kanten generell ineffizient ist. ♦

**Bemerkung** Leicht modifiziert eignen sich die Formate COO und CSR wie auch andere, um dünn besetzte Matrizen effizient abzuspeichern. Die Indices jedes Nichtnull-Eintrages wird als Kante in einem Digraphen verstanden, wobei nun der Kante der Werst des Eintrages zugeordnet wird. ♦

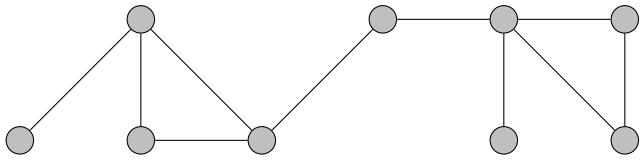
# 2 Etwas Graphentheorie

## 2.1 Zusammenhang

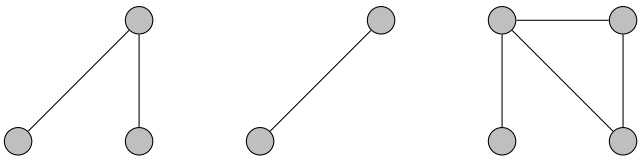
**Definition 2.1** Sei $G = (V, E)$ ein Graph und $C \subset V$. $C$ heisst *zusammenhängend*, falls je zwei Knoten $v, w \in C$, $v \neq w$, voneinander *erreichbar* sind, das heisst, falls gilt $w \in \text{post}^\star(v)$ (oder äquivalent $v \in \text{post}^\star(w)$). $C$ heisst *Zusammenhangskomponente* von $G$, falls $C$ eine nicht-leere maximale zusammenhängende Knotenmenge ist. Maximalität bedeutet hier, dass $C$ in keiner anderen zusammenhängenden Menge $C' \subset V$ echt enthalten ist (also $C \neq C'$). Der Graph $G$ heisst zusammenhängend, falls $V$ zusammenhängend ist.

**Beispiel 2.2**

- Ein zusammenhängender Graph ist:



- Ein unzusammenhängender Graph mit drei Zusammenhangskomponenten ist:



♣

Offenbar sind die Zusammenhangskomponenten eines Graphs die Äquivalenzklassen der Knotenmenge $V$ unter der Erreichbarkeits-Äquivalenzrelation "$\equiv$", wobei

$$v \equiv w \;:\Longleftrightarrow\; \{v\} \cup \text{post}^\star(v) = \{w\} \cup \text{post}^\star(w).$$

Insbesondere zerfällt $G$ in paarweise disjunkte Zusammenhangskomponenten $C_1, C_2, \ldots, C_r$ mit

$$V = \bigcup_{i=1}^{r} C_i, \qquad E = \bigcup_{i=1}^{r} E_i,$$

wobei $E_i := E \cap \{X \subseteq C_i \;:\; |X| = 2\}$.

**Satz 2.3** *Sei $G = (V, E)$ ein Graph mit $n = |V| \geq 1$ Knoten sowie $m = |E|$ Kanten, dann gilt: Aus $G$ zusammenhängend folgt $m \geq n - 1$.*

*Beweis.* Wir beweisen die Aussage durch Induktion nach $n$. Für $n = 1$ folgt $m = 0 = n - 1$; im Fall $n = 2$ ist $G$ zusammenhängend genau dann, wenn $m = 1 = n - 1$. Wir nehmen nun an, dass $n \geq 3$ und $G$ zusammenhängend ist. Wähle $v \in V$, so dass

$$\deg(v) = \min_{w \in V} \deg(w) =: k.$$

Es gilt $k > 0$, denn sonst wäre $v$ ein isolierter Knoten, was im Widerpruch zu $G$ zusammenhängend steht. Im Fall $k \geq 2$ folgt

$$2m = 2|E| = \sum_{w \in V} \underbrace{\deg(w)}_{\geq k} \geq n \cdot k \geq 2 \cdot n$$

und folglich $m \geq n \geq n - 1$.

Für $k = 1$ ergibt sich die Aussage wie folgt: Es sei $G' = (E', V')$ derjenige Graph, der durch Streichen des Knotens $v$ sowie der ausgehenden Kante entsteht. Mit $G$ ist auch $G'$ zusammenhängend und nach Induktionsvoraussetzung folgt wegen $|V'| = n-1$ und $|E'| = m-1$

$$m - 1 = |E'| \geq (n - 1) - 1 = n - 2,$$

das heisst $m \geq n - 1$. ♠

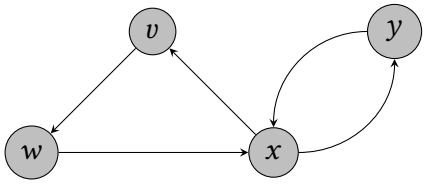
**Definition 2.4** Sei $G = (V, E)$ ein Digraph und $C \subset V$. Dann heisst $C$ zusammenhängend, falls $C$ in dem von $G$ induzierten Graphen zusammenhängend ist.

**Satz 2.5** *Sei $G = (V, E)$ ein Digraph mit $n = |V| \geq 1$ Knoten sowie $m = |E|$ Kanten, dann gilt: Aus $G$ zusammenhängend folgt $m \geq n - 1$.*

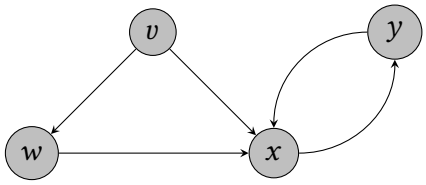
*Beweis.* Trivial. ♠

**Definition 2.6** Ein Digraph $G = (V, E)$ heisst *stark zusammenhängend*, falls für jedes Paar von Knoten $v, w \in V$ mit $v \neq w$ gilt $v \in \text{post}^\star(w)$ und $w \in \text{post}^\star(v)$, das heisst, es gibt einen Weg von $v$ nach $w$ und einen Weg von $w$ nach $v$. Die *starken Zusammenhangskomponenten* sind die maximalen stark zusammenhängenden Teilgraphen.

**Beispiel 2.7** Der Digraph



ist stark zusammenhängend. Hingegen besteht der nicht stark zusammenhängender Digraph




aus den starken Zusammenhangskomponenten $\{v\}, \{w\}, \{x, y\}$. ♣

Hinsichtlich der starken Zusammenhangskomponenten ist es oft nützlich, den so genannten *kondensierten Digraphen* zu betrachten. Dieser fasst die Knotenmengen der starken Zusammenhangskomponenten zu einzelnen Knoten zusammen.

**Definition 2.8** Für einen Digraphen $G = (V, E)$ seien die starken Zusammenhangskomponenten gegeben durch

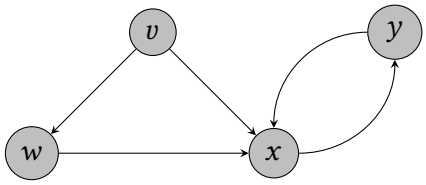
$$G_i := \big(V_i, E \cap (V_i \times V_i)\big) \quad \text{für } i = 1, 2, \dots, p$$

mit $V_1, V_2, \dots, V_p \subset V$. Der Graph $G^\star = (V^\star, E^\star)$ mit

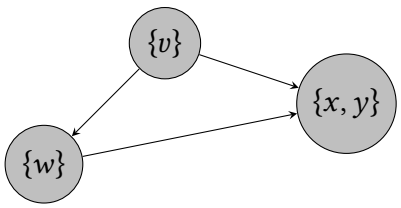
$$V^\star := \{V_1, V_2, \dots, V_p\} \quad \text{und} \quad E^\star := \Big\{(V_i, V_j) \in V^\star \times V^\star \,:\, i \neq j \text{ und } E \cap (V_i \times V_j) \neq \emptyset\Big\}$$

heisst *kondensierter Digraph* zu $G$.

**Beispiel 2.9** Der Digraph




hat den kondensierten Digraph




bestehend aus den starken Zusammenhangskomponenten $\{v\}, \{w\}, \{x, y\}$. ♣

## 2.2 Rundweg

**Definition 2.10** Ein *Eulerscher Rundweg* in einem Graphen oder Digraphen $G = (V, E)$ ist ein Rundweg, der jede Kante $e \in E$ genau einmal enthält. Ist $G$ ein Graph, so nennen wir $G$ *Eulersch*, falls der Grad jedes Knotens gerade ist. Ein Digraph ist *Eulersch*, falls $\text{indeg}(v) = \text{outdeg}(v)$ für alle $v \in V$ gilt.

Basierend auf dieser Definition haben wir den folgenden berühmten Satz, der Leonhard Euler zugeschrieben wird.

**Satz 2.11** *Ein zusammenhängender Graph oder Digraph $G = (V, E)$ besitzt genau dann einen Eulerschen Rundweg, wenn er Eulersch ist.*

*Beweis.* Die Bedingung ist notwendig, da ein Knoten $v \in V$, der $k$-mal in einem Eulerschen Rundweg vorkommt (oder $k + 1$ mal, wenn es sich um den Anfangs- und Endknoten handelt), im gerichteten Fall

$$\text{indeg}(v) = \text{outdeg}(v) = k$$

und im ungerichteten Fall

$$\deg(v) = 2k$$

erfüllen muss.

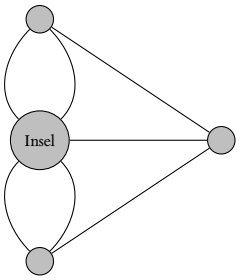
Dass die Bedingung auch hinreichend ist, sieht man wie folgt ein: Sei

$$\pi = v_0, v_1, \dots, v_r$$

der längste Weg, in dem jede Kante aus $E$ höchstens einmal vorkommt. Insbesondere muss in diesem Weg jede Kante enthalten sein, die $v_r$ verlässt. Das bedeutet aber sofort $v_0 = v_r$ wegen der Bedingung an den Knotengrad. Angenommen $\pi$ enthält nicht alle Kanten, das heisst, es gibt eine Kante $e = (w_1, w_2) \in E$ oder $e = \{w_1, w_2\} \in E$, sodass $e \neq (v_i, v_{i+1})$ beziehungsweise $e \neq \{v_i, v_{i+1}\}$ für alle $i = 0, \dots, r-1$. Da $G$ zusammenhängend ist, muss nun entweder $w_1$ oder $w_2$ in $\pi$ enthalten sein. Ist nun beispielsweise $w_1 = v_i$ in $\pi$ enthalten, so ist

$$\tilde{\pi} = v_i, v_{i+1}, \dots, v_r, v_1, \dots v_{i-1}, v_i, w_2$$

ein längerer Weg, in dem jede Kante genau einmal vorkommt. ♠



Zu Zeiten Eulers floss durch die Stadt Königsberg der Fluss Pregel. In diesem Fluss gab es eine Insel, hinter der sich der Fluss teilte. Die vier resultierenden Landstücke waren durch insgesamt sieben Brücken verbunden. Hieraus ergab sich die Fragestellung, ob es möglich wäre einen Rundweg zu finden, der jede Brücke nur genau einmal passiert. Die vorliegend Situation ist im vorangestellten *Multigraphen* (hierin sind auch parallele Kanten zugelassen) schematisch dargestellt. Der soeben bewiesene Satz sagt nun aus, dass in der vorliegenden Konfiguration kein solcher Rundweg existiert.

**Satz 2.12** *Sei $G = (V, E)$ ein Graph oder Digraph und*

$$\pi = v_0, v_1, \dots, v_r$$

*mit $r \geq 1$ ein Weg. Wenn $\pi$ nicht einfach ist, dann kann $\pi$ aus einem einfachen Weg gewonnen werden, indem wiederholt einfache Rundwege eingefügt werden. Insbesondere kann also jeder nicht einfache Rundweg daher aus einfachen Rundwegen zusammengesetzt werden.*

*Beweis.* Es ist einfach einzusehen, dass jeder Weg der Länge $\leq 2$ einfach sein muss. Daher reicht es per Induktion zu zeigen, dass ein nicht einfacher Weg aus einem Weg und einem Rundweg zusammengestzt werden kann, welche beide notwendigerweise eine echt kleinere Länge aufweisen. Sei also

$$\pi = v_0, v_1, \dots, v_r$$

mit $r \geq 1$ ein nicht einfacher Weg. Dann existieren $0 \leq i < j \leq r$, sodass $v_i = v_j$ und insbesondere auch $(i, j) \neq (0, r)$ gelten. Da wir keine Schleifen in Graphen und Digraphen erlauben, muss aber insbesondere auch $j > i + 1$ gelten.

- Ist $i > 0$, so besteht $\pi$ aus dem Weg

  $$\tilde{\pi} = v_0, \dots, v_i, v_{j+1}, \dots, v_r$$

  und dem Rundweg

  $$\rho = v_i, \dots, v_j.$$

- Andernfalls gelten $i = 0$ und $j \neq r$. Dann besteht $\pi$ aus dem Weg

  $$\tilde{\pi} = v_0, v_{j+1}, \dots, v_r$$

  und dem Rundweg

  $$\rho = v_0, \dots, v_j.$$

Da in beiden Fällen die Länge von $\tilde{\pi}$ und $\rho$ echt kleiner als die Länge von $\pi$ ist, sind wir fertig. ♠

*Beachte:* Ein nicht einfacher Weg ist im Allgemeinen nicht eindeutig aus einem einfachen Weg und einfachen Rundwegen zusammengesetzt.

**Beispiel 2.13** Der Rundweg $v_1, v_5, v_3, v_6, v_2, v_5, v_4, v_6, v_1$ lässt isch einerseits durch die Zusammensetzung der einfachen Rundwege $v_1, v_5, v_3, v_6, v_1$ und $v_6, v_2, v_5, v_4, v_6$ generieren, aber auch durch $v_1, v_5, v_4, v_6, v_1$ und $v_5, v_3, v_6, v_2, v_5$. ♣

## 2.3 Zyklus

**Definition 2.14** Ein einfacher Rundweg in einem Graphen $G = (V, E)$ heisst *pathologisch*, wenn er Länge 2 hat. Ein Rundweg in einem Graphen oder Digraphen heisst *pathologisch*, wenn jede Art den Rundweg aus einfachen Rundwegen zusammenzusetzen, mindestens einen pathologischen einfachen Rundweg benötigt. Wir nennen einen Rundweg, der nicht pathologisch ist, *Zyklus*.

*Beachte:* In der Literatur wird statt Zyklus auch *Kreis* benutzt.

**Bemerkung** Offensichtlich bedeutet diese Definition also:

(i) jeder nicht pathologische einfache Rundweg ist ein Zyklus,

(ii) sind $\pi_1 = v_0, v_1, \dots, v_r$ und $\pi_2 = w_0, w_1, \dots, w_\ell$ Zyklen mit $v_i = w_0 = w_\ell$, so ist auch

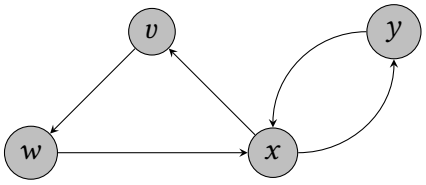
$$\pi = v_0, v_1, \dots, v_{i-1}, w_0, w_1, \dots, w_\ell, v_{i+1}, v_{i+2}, \dots, v_r$$

ein Zyklus,

(iii) nur die durch (i) und (ii) generierbaren Rundwege sind Zyklen. ♦

**Beispiel 2.15**

- Der Digraph



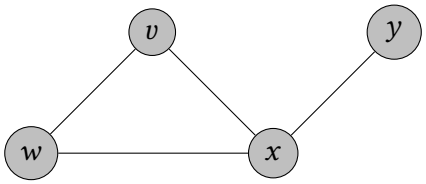

besitzt die einfachen Zyklen

$$\pi_1 = x, y, x, \qquad \pi_2 = v, w, x, v,$$

und die nicht einfachen Zyklen

$$\pi_3 = x, y, x, y, x, \qquad \pi_4 = v, w, x, y, x, v.$$

- Der Graph

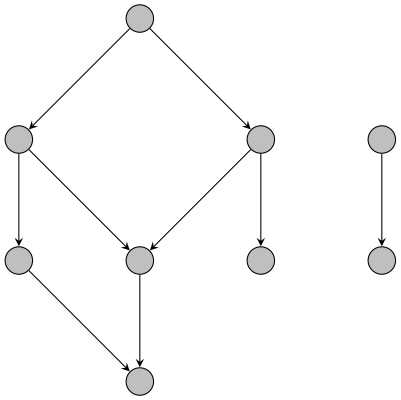



besitzt den (einfachen) Zyklus $\pi_1 = v, w, x, v$. Dementgegen sind die zwei Rundwege $\pi_2 = x, y, x$ und $\pi_3 = v, w, x, y, x, v$ keine Zyklen! ♣

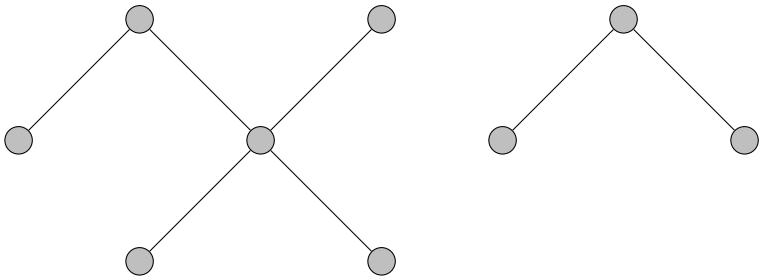
**Definition 2.16** Ein Graph oder Digraph $G$ heisst *azyklisch* oder *zyklenfrei*, falls es keine Zyklen in $G$ gibt. Ein azyklischer und zusammenhängender Graph ist ein *Baum*.

**Beispiel 2.17**

- Azyklischer Digraph:



- Azyklischer Graph:



♣

**Satz 2.18** *Sei $G = (V, E)$ ein Graph mit $n$ Knoten. Dann sind folgende Aussagen äquivalent:*

1. *$G$ ist ein Baum.*
2. *$G$ hat $n - 1$ Kanten und ist zusammenhängend.*

3. *$G$ hat $n-1$ Kanten und ist azyklisch.*
4. *$G$ ist azyklisch und das Hinzufügen einer beliebigen Kante erzeugt einen Zyklus.*
5. *$G$ ist zusammenhängend und das Entfernen einer Kante erzeugt einen unzusammenhängenden Graphen.*
6. *Jedes Paar von verschiedenen Knoten in $G$ ist durch genau einen einfachen Weg miteinander verbunden.*

*Beweis.*

$1 \Rightarrow 6$: Dies folgt aus der Tatsache, dass die Vereinigung zweier disjunkter einfacher Wege mit gleichen Anfangs- und Endpunkten ein Zyklus ist.

$6 \Rightarrow 5$: $G$ ist zusammenhängend gemäss Voraussetzung. Das Entfernen der Kante $\{v, w\}$ macht $w$ unerreichbar von $v$.

$5 \Rightarrow 4$: $G$ ist azyklisch, denn sonst kann eine Kante entfernt werden, so dass $G$ weiterhin zusammenhängend ist. Da es in $G$ stets einen Weg von $v$ nach $w$ gibt, liefert das Hinzufügen einer Kante $\{v, w\}$ einen Zyklus.

$4 \Rightarrow 3 \Rightarrow 2$: Die Behauptung folgt, falls für einen azyklischen, ungerichteten Graphen gilt

$$n = m + p, \tag{2.1}$$

wobei $m = |E|$ und $p$ die Anzahl der Zusammenhangskomponenten ist. Da (2.1) klar ist für $m = 0$, nehmen wir an, (2.1) gilt für ein $|E| = m$. Fügen wir eine zusätzliche Kante hinzu, dies bedeutet $|E| = m + 1$, so muss sich $p$ um eins reduzieren, denn sonst würde ein Zyklus entstehen.

$2 \Rightarrow 1$: Wir zerstören Zyklen aus $G$ durch Entfernen von Kanten. Haben wir etwa $k$ Kanten entfernt, so folgt aus (2.1)

$$\underbrace{n - 1 - k}_{\text{Kanten}} + \underbrace{p}_{=1} = n,$$

das heisst $k = 0$. ♠

# Index