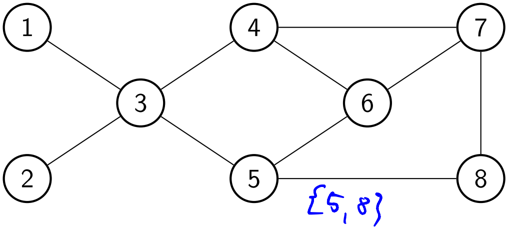
Woche 5: *23. Mai 2024*

Thema: *Graphentheorie*

5.1 Einleitung

Einfache, ungerichtete Graphen

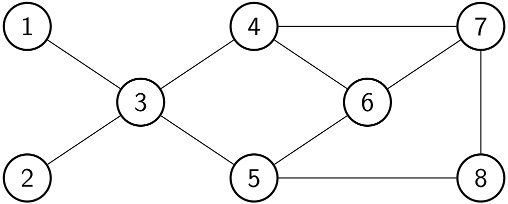


Definition. Ein einfacher, ungerichteter Graph G besteht aus einer Menge $V(G)$ von Knoten und einer Menge $E(G) \subseteq \mathcal{P}_2[V(G)]$ von Kanten.

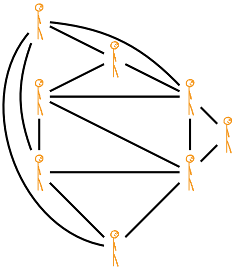
Anmerkung.

In der Literatur werden Knoten bisweilen auch Ecken genannt.

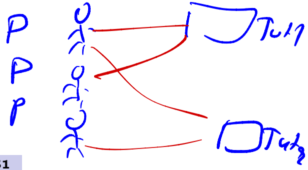
Einfache, ungerichtete Graphen



Beispiel. Die „sich-kennen Relation“ einer Gruppe von Personen modelliert als Graph.



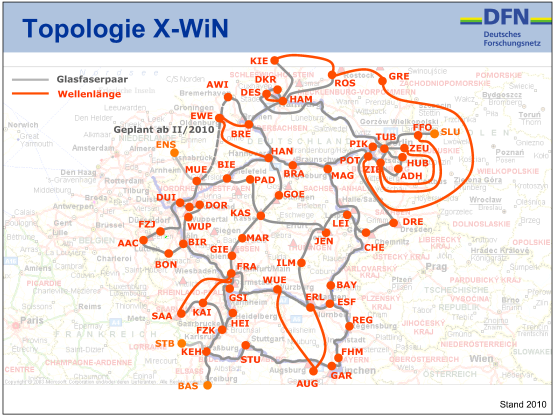
Definition. Ein einfacher, ungerichteter Graph G besteht aus einer Menge $V(G)$ von Knoten und einer Menge $E(G) \subseteq \mathcal{P}_2[V(G)]$ von Kanten.



Anmerkung.

In der Literatur werden Knoten bisweilen auch Ecken genannt.

Computer Netze als Graph

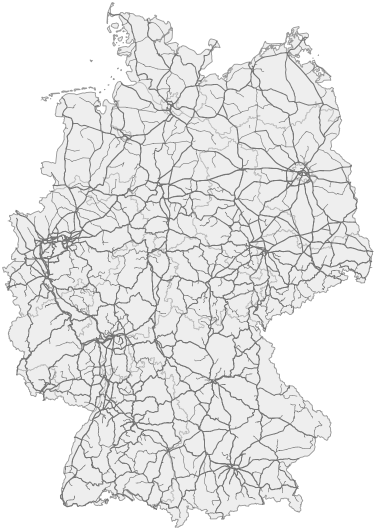


Computernetze.

Knoten: Computer/Router

Kanten: Verbindungen z.B. Glasfaser, Funk

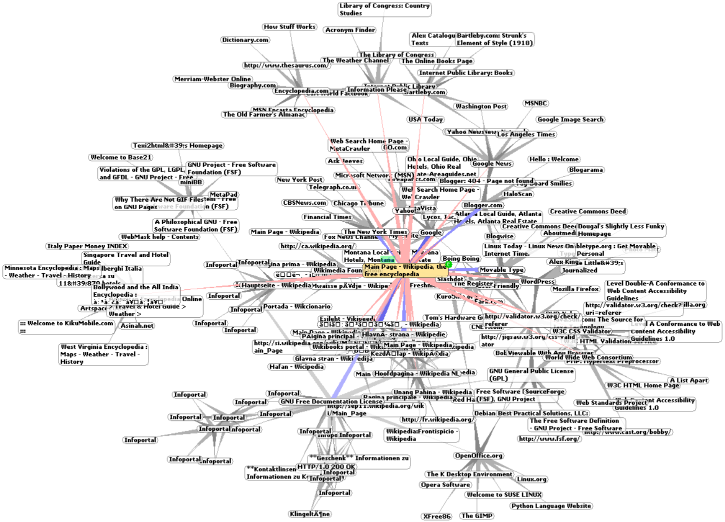
Straßen- und Bahnkarten



Schienennetze.

Knoten: Bahnhöfe
Kanten: Schienen

Das World Wide Web



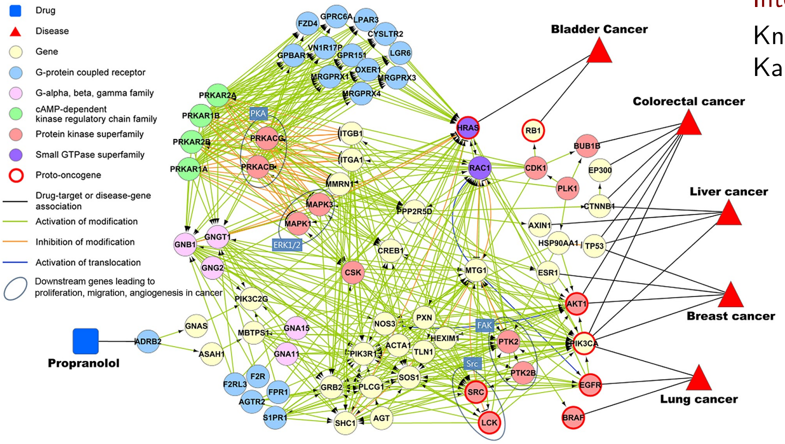
WWW.

Knoten: Webpages
Kanten: Links

Soziale Netzwerke.

Knoten: User
Kanten: Follows, Friends, ...

DNA-Protein Interaktionsnetzwerk



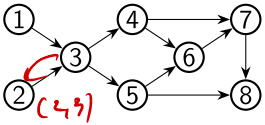
Interaktionsnetzwerke.

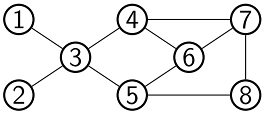
Knoten: Proteine, ...

Kanten: Interaktionen

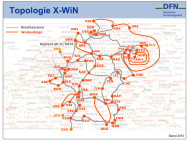
Arten von Graphen

Wir unterscheiden zwischen gerichteten und ungerichteten Graphen.

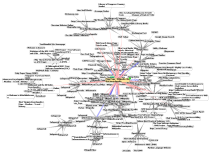




Ungerichtet. Computernetze



Gerichtet. Soziale Netze, WWW



Definition. Ein einfacher, ungerichteter Graph G besteht aus einer Menge $V(G)$ von Knoten und einer Menge $E(G) \subseteq \mathcal{P}_2[V(G)]$ von Kanten.

Definition. Ein gerichteter Graph G besteht aus einer Menge $V = V(G)$ von Knoten und einer Menge $E(G) \subseteq V \times V$ von Kanten.

Hinweis. Wir werden uns vor allem mit ungerichteten Graphen beschäftigen. Wenn nichts anderes gesagt wird, sind daher im folgenden alle Graphen ungerichtet.

5.2 Grundlagen

Grundbegriffe

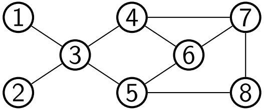
Definition.
einfacher, ungerichteter Graph G:
Knotenmenge $V(G)$
Kantenmenge $E(G) \subseteq \mathcal{P}_2[V(G)]$

Schleifen. Sei G ein Graph und $v \in V(G)$. Eine Kante von v zu sich selbst nennt man Schleife (oder self-loop).

Ein Graph heißt einfach, wenn er keine Schleifen enthält.

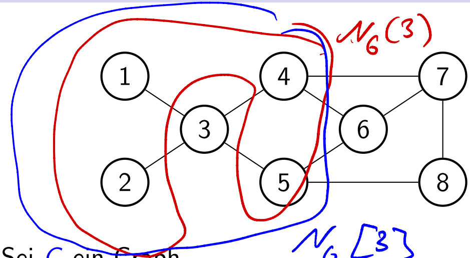
Definition.

- Die *Größe* $|G|$ eines Graphs G ist die Zahl der Knoten in G, also $|G| = |V(G)|$.
- Ein Graph G heißt *endlich*, wenn $|G|$ endlich ist.



Vereinbarung. Solange nichts anderes gesagt wird, sind alle Graphen in dieser Vorlesung endlich, einfach und ungerichtet.

Adjazenz, Inzidenz und Nachbarschaften



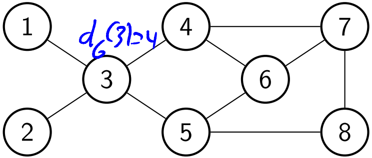
$N_G(3)$

$N_G[3]$

Definition. Sei G ein Graph.

- Zwei Knoten $u, v \in V(G)$ heißen *adjazent*, wenn $\{u, v\} \in E(G)$.
- Wenn $e = \{u, v\} \in E(G)$ eine Kante in G ist und u einer der Endpunkte von e, dann nennt man e und u *inzident*.
- Die *Nachbarschaft* $N_G(u)$ eines Knotens $u \in V(G)$ ist definiert als
$$N_G(u) := \{v \in V(G) : \{u, v\} \in E(G)\}.$$
- Die *geschlossene Nachbarschaft* $N_G[u]$ von $u \in V(G)$ ist definiert als $N_G[u] := N_G(u) \cup \{u\}$.

Grad eines Knoten



Definition. Sei G ein Graph.

- Wir definieren den *Grad* $d_G(v)$ von $v \in V(G)$ als $d_G(v) := |N_G(v)|$.

- Der *Maximalgrad* $\Delta(G)$ von G ist $\max\{d_G(v) : v \in V(G)\}$.

- Der *Minimalgrad* $\delta(G)$ von G ist $\min\{d_G(v) : v \in V(G)\}$.

Handshake Lemma

Satz.

In jedem Graph G mit mindestens zwei Knoten gibt es Knoten $u \neq v \in V(G)$ mit $d_G(u) = d_G(v)$.

Handshake Lemma

Satz.
In jedem Graph G mit mindestens zwei Knoten gibt es Knoten $u \neq v \in V(G)$ mit $d_G(u) = d_G(v)$.

Beweis. Die Aussage folgt direkt aus dem handshake lemma, das wir zu Beginn der Vorlesung bewiesen haben.

Lemma (handshake lemma).
Eine Menge M von Personen treffen sich auf einem Empfang. Einige davon geben sich zur Begrüßung gegenseitig die Hand.

Wenn $|M| > 1$ gibt es zwei Personen aus M, die genau gleich vielen Personen auf M die Hand geben.

Handshake Lemma

Satz.
In jedem Graph G mit mindestens zwei Knoten gibt es Knoten $u \neq v \in V(G)$ mit $d_G(u) = d_G(v)$.

Beweis. Die Aussage folgt direkt aus dem handshake lemma, das wir zu Beginn der Vorlesung bewiesen haben.

Dazu interpretieren wir die Knoten von G als „Personen“ und eine Kante zwischen u und v bedeutet, dass sich u und v per Handschlag begrüßen.

Lemma (handshake lemma).
Eine Menge M von Personen treffen sich auf einem Empfang. Einige davon geben sich zur Begrüßung gegenseitig die Hand.

Wenn $|M| > 1$ gibt es zwei Personen aus M, die genau gleich vielen Personen auf M die Hand geben.

Handshake Lemma

Satz.
In jedem Graph G mit mindestens zwei Knoten gibt es Knoten $u \neq v \in V(G)$ mit $d_G(u) = d_G(v)$.

Beweis. Die Aussage folgt direkt aus dem handshake lemma, das wir zu Beginn der Vorlesung bewiesen haben.

Dazu interpretieren wir die Knoten von G als „Personen“ und eine Kante zwischen u und v bedeutet, dass sich u und v per Handschlag begrüßen.

Der *Grad* eines Knotens u entspricht dann der Zahl der Personen, die u begrüßt.

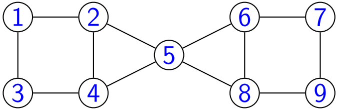
Lemma (handshake lemma).
Eine Menge M von Personen treffen sich auf einem Empfang. Einige davon geben sich zur Begrüßung gegenseitig die Hand.
Wenn $|M| > 1$ gibt es zwei Personen aus M, die genau gleich vielen Personen auf M die Hand geben.

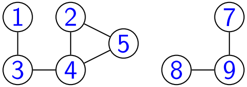
5.3 Untergraphen

Untergraphen

Definition. Seien G, H Graphen.

1. H ist ein Untergraph eines Graphs G, geschrieben $H \subseteq G$, wenn $V(H) \subseteq V(G)$ und $E(H) \subseteq E(G)$.



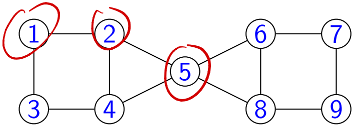


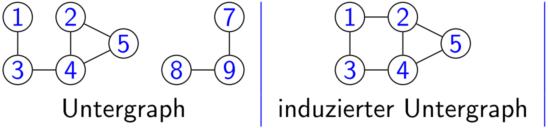
Untergraph

Untergraphen

Definition. Seien G, H Graphen.

1. H ist ein Untergraph eines Graphs G, geschrieben $H \subseteq G$, wenn $V(H) \subseteq V(G)$ und $E(H) \subseteq E(G)$.
2. H ist ein *induzierter Untergraph* von G, wenn $H \subseteq G$ und $E(H) = E(G) \cap \mathcal{P}_2(V(H))$.





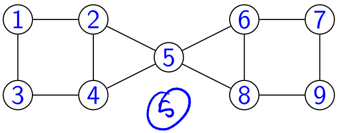
Untergraphen

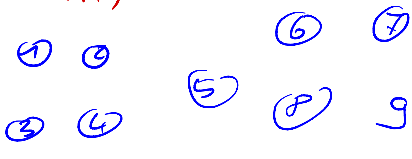
Definition. Seien G, H Graphen.

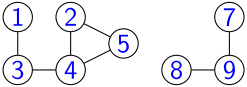
1. H ist ein Untergraph eines Graphs G, geschrieben $H \subseteq G$, wenn $V(H) \subseteq V(G)$ und $E(H) \subseteq E(G)$.
2. H ist ein *induzierter Untergraph* von G, wenn $H \subseteq G$ und $E(H) = E(G) \cap \mathcal{P}_2(V(H))$.
3. H ist ein *spannender Untergraph* von G, wenn $H \subseteq G$ und $V(H) = V(G)$.

2-elementige TM von V(H)

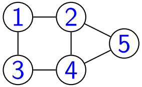
spann. U.G.



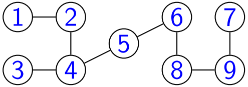




Untergraph



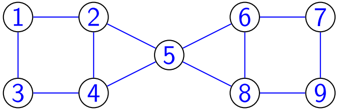
induzierter Untergraph



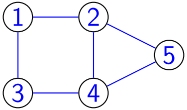
spannender Untergraph

Induzierte Untergraphen

Definition. Sei G ein Graph und $X \subseteq V(G)$. Der von X *in* G *induzierte Untergraph*, geschrieben $G[X]$, ist der induzierte Untergraph von G mit Knotenmenge $V(G[X]) = X$.



Graph G



$G[\{1, 2, 3, 4, 5\}]$

Induzierte Untergraphen

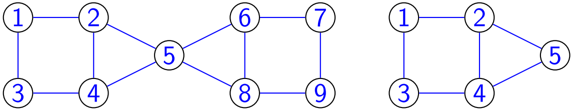
Definition. Sei G ein Graph und $X \subseteq V(G)$. Der von X *in* G *induzierte Untergraph*, geschrieben $G[X]$, ist der induzierte Untergraph von G mit Knotenmenge $V(G[X]) = X$.

Notation. Seien G, H Graphen. Mit $G - H$ bezeichnen wir den durch $V(G) \setminus V(H)$ in G induzierten Untergraphen.

Für $v \in V(G)$ definieren wir $G - v$ als $G[V(G) \setminus \{v\}]$.

Analog definieren wir $G - S$ für eine Menge $S \subseteq V(G)$.

Für eine Kante $e \in E(G)$ definieren wir $G - e$ als den Graphen mit der Knotenmenge $V(G)$ und Kantenmenge $E(G) \setminus \{e\}$.



Graph G $\quad$ $G[\{1, 2, 3, 4, 5\}]$

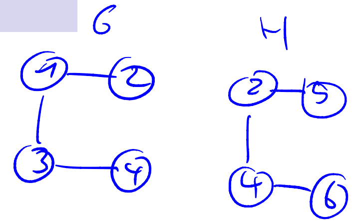
Vereinigung und Schnitt

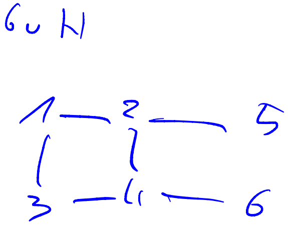
Definition.

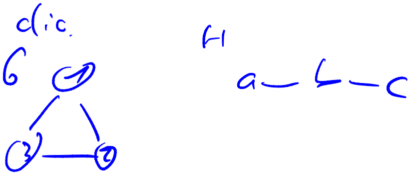
Mit $G \cup H$ bezeichnen wir die Vereinigung der Graphen G und H,

d.h. den Graph mit Knotenmenge $V(G) \cup V(H)$ und Kantenmenge $E(G) \cup E(H)$.

Wenn $V(H) \cap V(G) = \emptyset$, dann nennen wir $G \cup H$ die disjunkte Vereinigung von G und H.







Vereinigung und Schnitt

Definition.

Mit $G \cup H$ bezeichnen wir die Vereinigung der Graphen G und H,

d.h. den Graph mit Knotenmenge $V(G) \cup V(H)$ und Kantenmenge $E(G) \cup E(H)$.

Wenn $V(H) \cap V(G) = \emptyset$, dann nennen wir $G \cup H$ die disjunkte Vereinigung von G und H.

Entsprechend definieren wir den Schnitt $G \cap H$ als den Graph mit Knotenmenge $V(G) \cap V(H)$ und Kantenmenge $E(G) \cap E(H)$.

5.4 Wege, Pfade und Zusammenhang

Pfade, Wege und Kreise

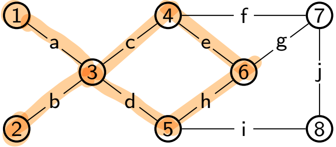
Definition. Sei G ein Graph.

1. Ein Weg W, oder auch *Kantenzug*, in G ist eine Folge

$$W = v_0 e_1 v_1 e_2 \dots v_{k-1} e_k v_k,$$

für ein $k \geq 0$, wobei $v_i \in V(G)$ für alle $0 \leq i \leq k$ und $e_i = \{v_{i-1}, v_i\} \in E(G)$ für alle $1 \leq i \leq k$.

Beispiel.



$W = (1, a, 3, d, 5, h, 6, e, 4, c, 3, b, 2)$

W ist ein Weg von 1 nach 2.

W verbindet 1 und 2.

Pfade, Wege und Kreise

Definition. Sei G ein Graph.

1. Ein Weg W, oder auch *Kantenzug*, in G ist eine Folge

$$W = v_0 e_1 v_1 e_2 \dots v_{k-1} e_k v_k,$$

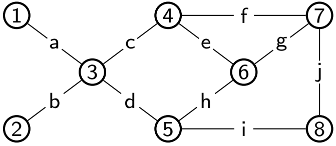
für ein $k \geq 0$, wobei $v_i \in V(G)$ für alle $0 \leq i \leq k$ und $e_i = \{v_{i-1}, v_i\} \in E(G)$ für alle $1 \leq i \leq k$.

Notation. In einfachen Graphen identifizieren wir W meistens mit der Folge $v_0 \dots v_k$ von Knoten, die W eindeutig bestimmt.

Die Knoten $v_1, \dots, v_{k-1}$ heißen innere Knoten, die Knoten v_0 und v_k sind die Endknoten von W.

W verbindet v_0 und v_k.

Beispiel.



$W = (1, a, 3, d, 5, h, 6, e, 4, c, 3, b, 2)$

W ist ein Weg von 1 nach 2.

W verbindet 1 und 2.

Pfade, Wege und Kreise

Definition. Sei G ein Graph.

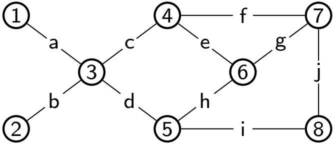
1. Ein Weg W, oder auch *Kantenzug*, in G ist eine Folge

$$W = v_0 e_1 v_1 e_2 \dots v_{k-1} e_k v_k,$$

für ein $k \geq 0$, wobei $v_i \in V(G)$ für alle $0 \leq i \leq k$ und $e_i = \{v_{i-1}, v_i\} \in E(G)$ für alle $1 \leq i \leq k$.

2. Die *Länge* von W ist die Zahl k der Kanten auf W.

Beispiel.



$W = (1, a, 3, d, 5, h, 6, e, 4, c, 3, b, 2)$

W ist ein Weg der Länge 5 von 1 nach 2.

W verbindet 1 und 2.

Pfade, Wege und Kreise

Definition. Sei G ein Graph.

1. Ein Weg W, oder auch *Kantenzug*, in G ist eine Folge

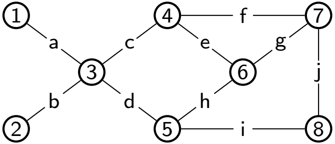
$$W = v_0 e_1 v_1 e_2 \dots v_{k-1} e_k v_k,$$

für ein $k \geq 0$, wobei $v_i \in V(G)$ für alle $0 \leq i \leq k$ und $e_i = \{v_{i-1}, v_i\} \in E(G)$ für alle $1 \leq i \leq k$.

2. Die *Länge* von W ist die Zahl k der Kanten auf W.

3. Ein *geschlossener Weg* ist ein Weg W dessen Endknoten übereinstimmen.

Beispiel.



$W = (1, a, 3, d, 5, h, 6, e, 4, c, 3, b, 2)$

W ist ein Weg der Länge 5 von 1 nach 2.

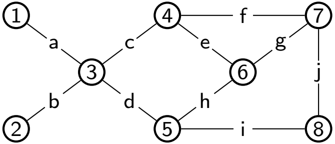
W verbindet 1 und 2.

Pfade, Wege und Kreise

Definition. Sei G ein Graph.

1. Ein *Weg* W, oder auch *Kantenzug*, in G ist eine Folge
$$W = v_0 e_1 v_1 e_2 \dots v_{k-1} e_k v_k,$$
für ein $k \geq 0$, wobei $v_i \in V(G)$ für alle $0 \leq i \leq k$ und $e_i = \{v_{i-1}, v_i\} \in E(G)$ für alle $1 \leq i \leq k$.
2. Die *Länge* von W ist die Zahl k der Kanten auf W.
3. Ein *geschlossener Weg* ist ein Weg W dessen Endknoten übereinstimmen.
4. Ein *Pfad* P in G ist ein Weg $v_0 \dots v_k$ mit $v_i \neq v_j$ für alle $0 \leq i < j \leq k$.

Beispiel.



$W = (1, a, 3, d, 5, h, 6, e, 4, c, 3, b, 2)$

W ist ein Weg der Länge 5 von 1 nach 2.

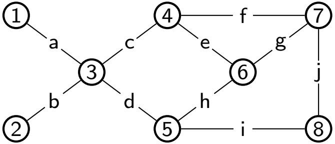
W verbindet 1 und 2.

Pfade, Wege und Kreise

Definition. Sei G ein Graph.

1. Ein Weg W, oder auch *Kantenzug*, in G ist eine Folge
$$W = v_0 e_1 v_1 e_2 \dots v_{k-1} e_k v_k,$$
für ein $k \geq 0$, wobei $v_i \in V(G)$ für alle $0 \leq i \leq k$ und $e_i = \{v_{i-1}, v_i\} \in E(G)$ für alle $1 \leq i \leq k$.
2. Die *Länge* von W ist die Zahl k der Kanten auf W.
3. Ein *geschlossener Weg* ist ein Weg W dessen Endknoten übereinstimmen.
4. Ein Pfad P in G ist ein Weg $v_0 \dots v_k$ mit $v_i \neq v_j$ für alle $0 \leq i < j \leq k$.
5. Ein Zyklus, oder auch Kreis, in G ist ein geschlossener Weg $v_0 e_1 v_1 \dots v_{k-1} e_k v_k e_{k+1} v_0$, für ein $k \geq 2$, so dass $v_0 e_1 \dots e_k v_k$ ein Pfad ist.

Beispiel.



$W = (1, a, 3, d, 5, h, 6, e, 4, c, 3, b, 2)$

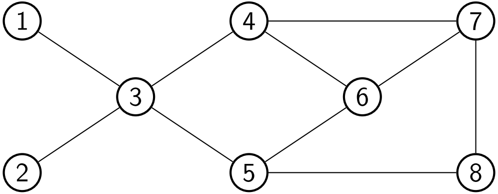
W ist ein Weg der Länge 5 von 1 nach 2.

W verbindet 1 und 2.

Zusammenhang in Graphen

Definition. Sei G ein nicht-leerer Graph.

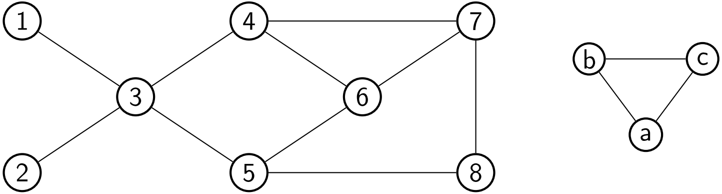
1. G ist zusammenhängend, wenn es zwischen je zwei Knoten $u, v \in V(G)$ einen Pfad in G gibt.
2. Wir nennen $U \subseteq V(G)$ *zusammenhängend in* G, wenn $G[U]$ zusammenhängend ist.



Zusammenhang in Graphen

Definition. Sei G ein nicht-leerer Graph.

1. G ist zusammenhängend, wenn es zwischen je zwei Knoten $u, v \in V(G)$ einen Pfad in G gibt.
2. Wir nennen $U \subseteq V(G)$ *zusammenhängend in G*, wenn $G[U]$ zusammenhängend ist.
3. Die maximalen zusammenhängenden Untergraphen von G heißen die Zusammenhangskomponenten von G.



5.5 Bäume und Wälder

Ein klein wenig Botanik

Definition.

1. Ein Graph G heißt *azyklisch*, wenn er keinen Kreis enthält.
2. Ein *Wald* ist ein azyklischer Graph G.

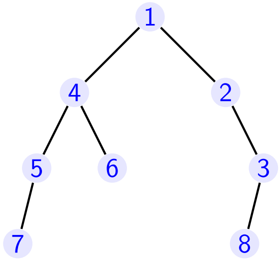
Ein klein wenig Botanik

Definition.

1. Ein Graph G heißt *azyklisch*, wenn er keinen Kreis enthält.
2. Ein *Wald* ist ein azyklischer Graph G.
3. Ein *Baum* ist ein zusammenhängender Wald, d.h. ein zusammenhängender, azyklischer Graph.


Minimalgrad in Bäumen

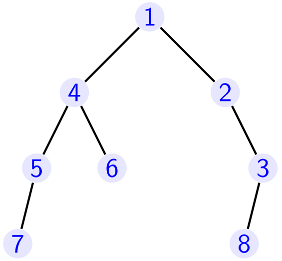
Lemma.
Jeder endliche Baum T hat Minimalgrad $\delta(T) \leq 1$.



Minimalgrad in Bäumen

Lemma.
Jeder endliche Baum T hat Minimalgrad $\delta(T) \leq 1$.

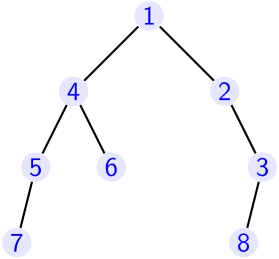
Beweis (durch Widerspruch). Angenommen, $\delta(T) > 1$.



Minimalgrad in Bäumen

Lemma.
Jeder endliche Baum T hat Minimalgrad $\delta(T) \leq 1$.

Beweis (durch Widerspruch). Angenommen, $\delta(T) > 1$.
Dann hat T mindestens 3 Knoten und 1 Kante.



Minimalgrad in Bäumen

Lemma.

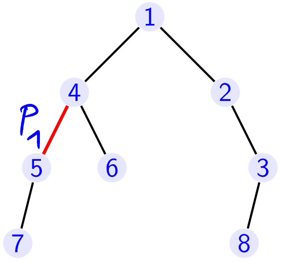
Jeder endliche Baum T hat Minimalgrad $\delta(T) \leq 1$.

Beweis (durch Widerspruch). Angenommen, $\delta(T) > 1$.

Dann hat T mindestens 3 Knoten und 1 Kante.

Wir konstruieren induktiv einen Pfad $P_i := u_0 \dots u_i$ wie folgt.

IA. Wähle $\{u_0, u_1\} \in E(G)$ beliebig. Wir definieren $P_1 := u_0 u_1$.



Minimalgrad in Bäumen

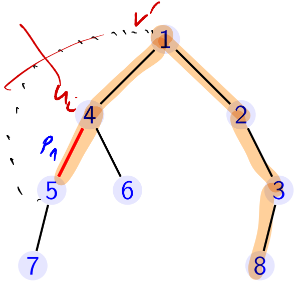
Lemma.
Jeder endliche Baum T hat Minimalgrad $\delta(T) \leq 1$.

Beweis (durch Widerspruch). Angenommen, $\delta(T) > 1$.
Dann hat T mindestens 3 Knoten und 1 Kante.
Wir konstruieren induktiv einen Pfad $P_i := u_0 \dots u_i$ wie folgt.

IA. Wähle $\{u_0, u_1\} \in E(G)$ beliebig. Wir definieren $P_1 := u_0 u_1$.

IV. Sei $P_i := u_0 \dots u_i$ schon definiert.

IS. Betrachte u_i. Da $\delta(G) > 1$, gibt es $v_1 \neq v_2 \in V(G)$ mit $\{u_i, v_1\}, \{u_i, v_2\} \in E(G)$.
D.h., es gibt ein $v' \in V(G)$ mit $v' \neq u_{i-1}$ und $\{u_i, v'\} \in E(G)$.

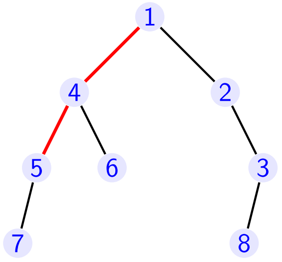


Minimalgrad in Bäumen

Lemma.
Jeder endliche Baum T hat Minimalgrad $\delta(T) \leq 1$.

Beweis (durch Widerspruch). Angenommen, $\delta(T) > 1$.
Dann hat T mindestens 3 Knoten und 1 Kante.
Wir konstruieren induktiv einen Pfad $P_i := u_0 \dots u_i$ wie folgt.

- IA. Wähle $\{u_0, u_1\} \in E(G)$ beliebig. Wir definieren $P_1 := u_0 u_1$.
- IV. Sei $P_i := u_0 \dots u_i$ schon definiert.
- IS. Betrachte u_i. Da $\delta(G) > 1$, gibt es $v_1 \neq v_2 \in V(G)$ mit $\{u_i, v_1\}, \{u_i, v_2\} \in E(G)$.
  D.h., es gibt ein $v' \in V(G)$ mit $v' \neq u_{i-1}$ und $\{u_i, v'\} \in E(G)$.
  Wir definieren $P_{i+1} := u_0 \dots u_i v'$. Das schließt die Induktion ab.



Minimalgrad in Bäumen

Lemma.
Jeder endliche Baum T hat Minimalgrad $\delta(T) \leq 1$.

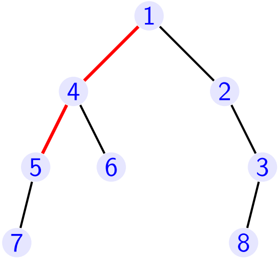
Beweis (durch Widerspruch). Angenommen, $\delta(T) > 1$.

Dann hat T mindestens 3 Knoten und 1 Kante.

Wir konstruieren induktiv einen Pfad $P_i := u_0 \dots u_i$ wie folgt.

- IA. Wähle $\{u_0, u_1\} \in E(G)$ beliebig. Wir definieren $P_1 := u_0 u_1$.
- IV. Sei $P_i := u_0 \dots u_i$ schon definiert.
- IS. Betrachte u_i. Da $\delta(G) > 1$, gibt es $v_1 \neq v_2 \in V(G)$ mit $\{u_i, v_1\}, \{u_i, v_2\} \in E(G)$.
  D.h., es gibt ein $v' \in V(G)$ mit $v' \neq u_{i-1}$ und $\{u_i, v'\} \in E(G)$.
  Wir definieren $P_{i+1} := u_0 \dots u_i v'$. Das schließt die Induktion ab.

Wir betrachten nun $P := P_{|T|+1}$. Da T azyklisch ist, ist P ein Pfad.



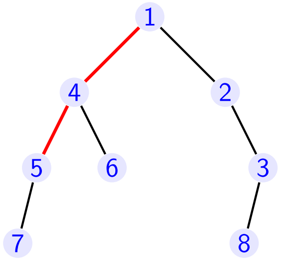
Minimalgrad in Bäumen

Lemma.
Jeder endliche Baum T hat Minimalgrad $\delta(T) \leq 1$.

Beweis (durch Widerspruch). Angenommen, $\delta(T) > 1$.
Dann hat T mindestens 3 Knoten und 1 Kante.
Wir konstruieren induktiv einen Pfad $P_i := u_0 \dots u_i$ wie folgt.

- IA. Wähle $\{u_0, u_1\} \in E(G)$ beliebig. Wir definieren $P_1 := u_0 u_1$.
- IV. Sei $P_i := u_0 \dots u_i$ schon definiert.
- IS. Betrachte u_i. Da $\delta(G) > 1$, gibt es $v_1 \neq v_2 \in V(G)$ mit $\{u_i, v_1\}, \{u_i, v_2\} \in E(G)$.
  D.h., es gibt ein $v' \in V(G)$ mit $v' \neq u_{i-1}$ und $\{u_i, v'\} \in E(G)$.
  Wir definieren $P_{i+1} := u_0 \dots u_i v'$. Das schließt die Induktion ab.

Wir betrachten nun $P := P_{|T|+1}$. Da T azyklisch ist, ist P ein Pfad.
Aber P enthält $|T| + 2$ verschiedene Knoten, was uns den gewünschten Widerspruch liefert. □



Zahl der Kanten eines Baums

Satz. Ein Baum T mit $n > 0$ Knoten hat genau $n - 1$ Kanten.

Zahl der Kanten eines Baums

Satz. Ein Baum T mit $n > 0$ Knoten hat genau $n - 1$ Kanten.

Beweis. (per Induktion über n)

(IA). Für $n = 1$ hat T einen Knoten und keine Kante.

(IV). Angenommen, die Behauptung gilt für alle $j < n$.

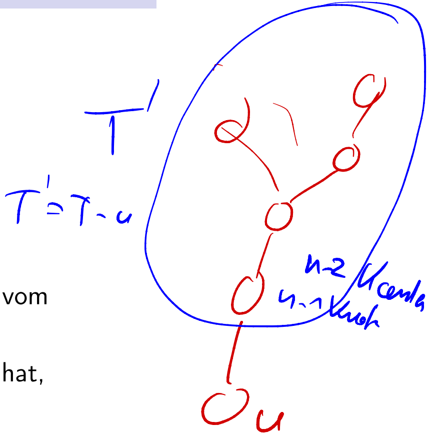
(IS). Es gilt $|V(T)| = n > 1$.

Nach dem letzten Lemma hat T ein Knoten $u \in V(T)$ vom Grad 1. Sei v der Nachbar von u in T.

Dann ist $T' := T - u$ ein Baum mit $n - 1$ Knoten und hat, nach (IV), $n - 2$ Kanten.

T hat aber einen Knoten u und eine Kante $\{u, v\}$ mehr als T'.

Also hat T genau $n - 1$ Kanten. □



5.6 Spannbäume

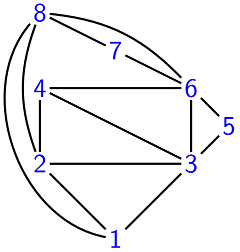
Spannbäume

Definition.

Ein *Spannbaum* eines Graphs G ist ein spannender, zusammenhängender und azyklischer Untergraph $T \subseteq G$.

Ein Spannbaum ist also ein Baum, der genau die Knoten von G enthält.

Beispiel.



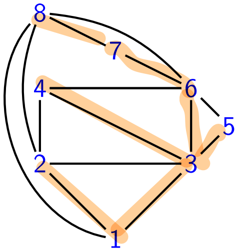
Spannbäume

Definition.
Ein *Spannbaum* eines Graphs G ist ein spannender, zusammenhängender und azyklischer Untergraph $T \subseteq G$.

Ein Spannbaum ist also ein Baum, der genau die Knoten von G enthält.

Lemma.
Jeder zusammenhängende Graph G enthält einen Spannbaum.

Beispiel.

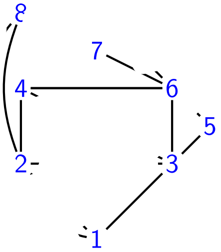


Spannbäume

Lemma.

Jeder zusammenhängende Graph G enthält einen Spannbaum.

Beispiel.



Spannbäume

Lemma.
Jeder zusammenhängende Graph G enthält einen Spannbaum.

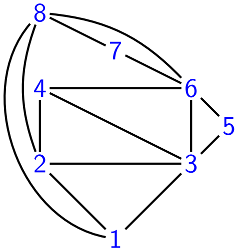
Beweis. Angenommen, die Behauptung sei falsch.

Dann gibt es also einen zusammenhängenden Graph, der keinen Spannbaum enthält.

Unter allen solchen Graphen wählen wir ein Graph $G = (V, E)$ mit minimaler Anzahl an Kanten.

Beobachtung. $|V| > 2$, da sonst G selbst ein Baum ist.

Beispiel.



Spannbäume

Lemma.
Jeder zusammenhängende Graph G enthält einen Spannbaum.

Beweis. Angenommen, die Behauptung sei falsch.

Dann gibt es also einen zusammenhängenden Graph, der keinen Spannbaum enthält.

Unter allen solchen Graphen wählen wir ein Graph $G = (V, E)$ mit minimaler Anzahl an Kanten.

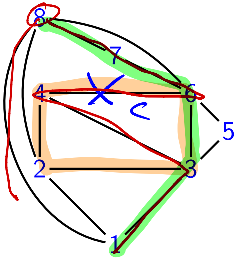
Beobachtung. $|V| > 2$, da sonst G selbst ein Baum ist.

Da G nach Voraussetzung zusammenhängend aber kein Baum ist, muss G einen Kreis $C = (v_1, e_1, v_2, \dots, v_k)$ enthalten.

Behauptung. $G - e_1$ ist zusammenhängend.

Beweis. Jeder Pfad zwischen zwei Knoten, der die Kante e_1 enthält, kann entlang $C - e_1$ „umgeleitet“ werden. $\dashv$

Beispiel.



Spannbäume

Lemma.
Jeder zusammenhängende Graph G enthält einen Spannbaum.

Beweis. Angenommen, die Behauptung sei falsch.

Dann gibt es also einen zusammenhängenden Graph, der keinen Spannbaum enthält.

Unter allen solchen Graphen wählen wir ein Graph $G = (V, E)$ mit minimaler Anzahl an Kanten.

Beobachtung. $|V| > 2$, da sonst G selbst ein Baum ist.

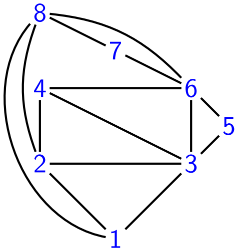
Da G nach Voraussetzung zusammenhängend aber kein Baum ist, muss G einen Kreis $C = (v_1, e_1, v_2, \dots, v_k)$ enthalten.

Behauptung. $G - e_1$ ist zusammenhängend.

Beweis. Jeder Pfad zwischen zwei Knoten, der die Kante e_1 enthält, kann entlang $C - e_1$ „umgeleitet“ werden. $\dashv$

Also ist $G - e_1$ zusammenhängend und hat keinen Spannbaum, hat aber weniger Kanten als G, im Widerspruch zur Wahl von G. $\square$

Beispiel.



Routing in Spannbäumen

Spannbäume.

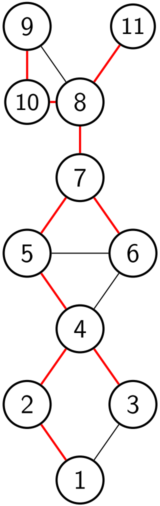
Sei G ein zshg. Graph und T ein Spannbaum von G. Offensichtlich ist

- jeder Pfad zwischen zwei Knoten u, v in T auch ein Pfad von u nach v in G und
- je zwei Knoten u und v sind in T durch einen Pfad verbunden.

Spannbäume sind eine kantenminimale Art, alle Knoten eines Graphen zu verbinden.

So etwas kann z.B. im Aufbau eines Netzwerks interessant sein, wenn Verbindungen (also Kanten) teuer zu bauen sind.

Allerdings kann der Pfad in T viel länger als der kürzeste Pfad in G sein.



5.7 Tiefen- und Breitensuche

Suchen in Graphen

Spannbäume. Der vorherige Beweis für die Existenz von Spannbäumen ist rein existenziell, d.h. er beweist, dass jeder zusammenhängende Graph einen Spannbaum hat, liefert aber keinen Algorithmus um einen solchen zu kontruieren.

Kontruktive Verfahren. Wir werden jetzt zwei Methoden kennenlernen, die Tiefen- und die Breitensuche, um Graphen systematisch durchsuchen zu können.

Beide Methoden liefern als „Nebeneffekt" einen Spannbaum in zusammenhängenden Graphen.

Tiefensuche

Sei G ein Graph mit $n > 0$ Knoten und sei $v \in V(G)$.

Für $1 \leq i \leq n$ konstruieren wir

- einen Unterbaum $T_i \subseteq G$ mit Wurzel v und
- eine Beschriftungsfunktion $\mathcal{I}_i : V(T_i) \to \{1, \dots, i\}$:

(IA). Definiere $T_1 := (\{v\}, \emptyset)$ und $\mathcal{I}_1(v) := 1$.

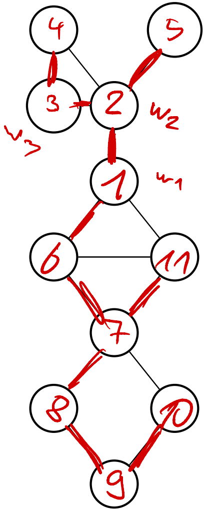
(IV). Seien $T_i, \mathcal{I}_i$ schon definiert.

(IS). Für $1 \leq j \leq i$ sei $\mathcal{I}_i(j) := w_j$.

Sei $m := \max\{j : N_G(w_j) \setminus V(T_i) \neq \emptyset\}$.

Wähle $u \in N_G(w_m) \setminus V(T_i)$ und definiere

$$
\begin{aligned}
V(T_{i+1}) &= V(T_i) \cup \{u\} \\
E(T_{i+1}) &= E(T_i) \cup \{\{w_m, u\}\} \\
\mathcal{I}_{i+1}(u) &= i + 1 \text{ und } \mathcal{I}_{i+1}(w_j) = j = I_i(w_j) \text{ für } 1 \leq j \leq i.
\end{aligned}
$$



Breitensuche

Sei G ein Graph mit $n > 0$ Knoten und sei $v \in V(G)$.

Für $1 \leq i \leq n$ konstruieren wir

- einen Unterbaum $T_i \subseteq G$ mit Wurzel v und
- eine Beschriftungsfunktion $\mathcal{I}_i : V(T_i) \to \{1, \dots, i\}$:

(IA). Definiere $T_1 := (\{v\}, \emptyset)$ und $\mathcal{I}_1(v) := 1$.

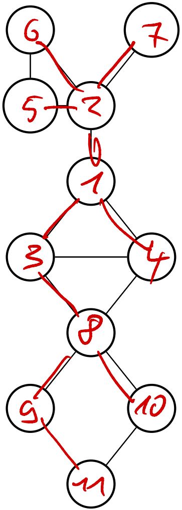
(IV). Seien $T_i, \mathcal{I}_i$ schon definiert.

(IS). Für $1 \leq j \leq i$ sei $\mathcal{I}_i(j) := w_j$.

Sei $m := \min\{j : N_G(w_j) \setminus V(T_i) \neq \emptyset\}$.

Wähle $u \in N_G(w_m) \setminus V(T_i)$ und definiere

$$
\begin{aligned}
V(T_{i+1}) &= V(T_i) \cup \{u\} \\
E(T_{i+1}) &= E(T_i) \cup \{\{w_m, u\}\} \\
\mathcal{I}_{i+1}(u) &= i + 1 \text{ und } \mathcal{I}_{i+1}(w_j) = j = l_i(w_j) \text{ für } 1 \leq j \leq i.
\end{aligned}
$$



Unterschied Tiefen- und Breitensuche

Tiefen- und Breitensuche. Beide Verfahren durchlaufen jeden Knoten eines zshg. Baums und generieren einen Spannbaum.

Breitensuche.

Sei T_v der durch eine Breitensuche in G von v entstandende Spannbaum.

Für $u \in V(G)$ ist der Pfad von v nach u in T ein kürzester Pfad von u nach v in G.

Der Vorteil der Breitensuche ist, dass sie Spannbäume mit kürzesten Pfaden von der Wurzel zu jedem Knoten aus $V(G)$ konstruiert.

Tiefensuche.

Die Tiefensuche konstruiert keine kürzesten Pfade, hat aber ansonsten viele Vorteile.

5.8 Zusammenhang und Erreichbarkeit in Graphen

Distanzen in Graphen.

Definition. Sei G ein Graph und $u \neq v \in V(G)$.

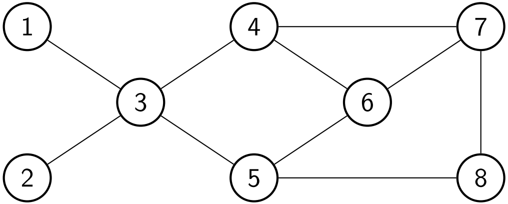
1. Die *Distanz* $dist_G(u,v)$ von u und v in G ist die minimale Länge eines Pfades von u nach v. Formal:

   $$dist_G(u,v) := \min\{\ell : \text{es ex. Pfad der Länge } \ell \text{ von } u \text{ nach } v \text{ in } G\}.$$

2. Ein *kürzester Pfad* zwischen u und v ist ein Pfad P von u nach v der Länge $dist_G(u,v)$.

Beispiel. $dist_G(1,7) = 3$.

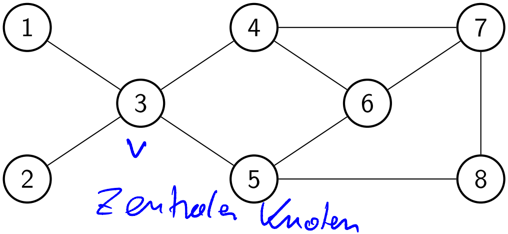
Ein kürzester Pfad zwischen 1 und 7 ist z.B. $1\,3\,4\,7$.



Distanzen in Graphen.

Definition. Sei G ein Graph und $u \neq v \in V(G)$.

1. Der *Durchmesser* von G ist $\max\{dist_G(u,v) : u,v \in V(G)\}$.
2. Der *Radius* von G ist $\min_{v \in V(G)} \max\{dist_G(u,v) : u \in V(G)\}$.



Durchmesser: 3

Radius: 2

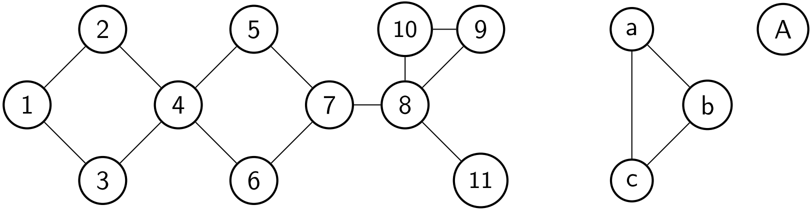
- **半径**（Radius）：图中所有节点中，从一个节点到其它所有节点最短路径中的最大值中的最小值。这里的半径为2，意味着至少有一个节点，它到图中任意其他节点的最远距离是2。

Zusammenhangskomponenten in Graphen

Erinnerung. Sei G ein nicht-leerer Graph.

Eine *Zusammenhangskomponente* von G ist ein maximaler (in Bezug auf die Untergraphrelation), zusammenhängender Untergraph $C \subseteq G$.

Beispiel.

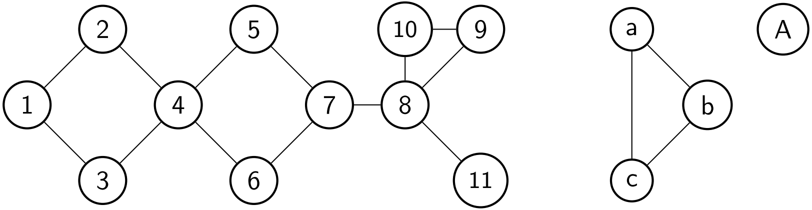


Zusammenhangskomponenten in Graphen

Erinnerung. Sei G ein nicht-leerer Graph.

Eine *Zusammenhangskomponente* von G ist ein maximaler (in Bezug auf die Untergraphrelation), zusammenhängender Untergraph $C \subseteq G$.

Beispiel.



Die Zusammenhangskomponenten sind die durch die Mengen

$$V_1 := \{1, \dots, 11\}, \qquad V_2 := \{a, b, c\} \qquad V_3 := \{A\}$$

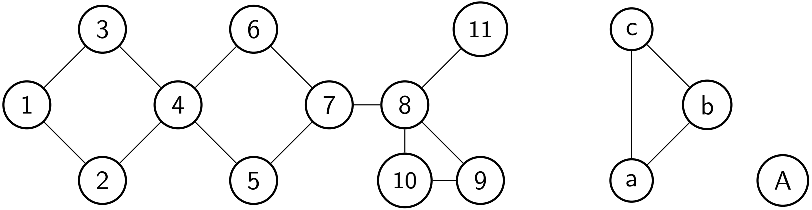
induzierten Untergraphen.

Zusammenhangskomponenten in Graphen

Berechnen von Komponenten. Wir können Zshg.-Komponenten durch Tiefen- oder Breitensuche ausrechnen.

1. Wähle $v \in V(G)$ beliebig.
2. Berechne einen DFS-Baum T_v mit Wurzel v.
   Dann ist $G[V(T_v)]$ eine Komponente von G.
3. Wenn es noch unmarkierte Knoten gibt, wähle einen solchen Knoten v und wiederhole Schritt 2.

Beispiel.

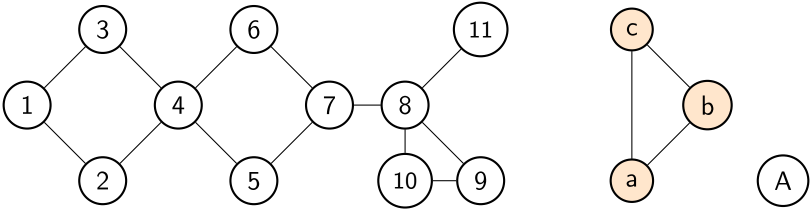


Zusammenhangskomponenten in Graphen

Berechnen von Komponenten. Wir können Zshg.-Komponenten durch Tiefen- oder Breitensuche ausrechnen.

1. Wähle $v \in V(G)$ beliebig.
2. Berechne einen DFS-Baum T_v mit Wurzel v.
   Dann ist $G[V(T_v)]$ eine Komponente von G.
3. Wenn es noch unmarkierte Knoten gibt, wähle einen solchen Knoten v und wiederhole Schritt 2.

Beispiel.

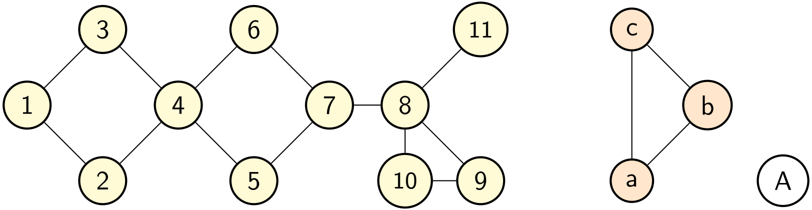


Zusammenhangskomponenten in Graphen

Berechnen von Komponenten. Wir können Zshg.-Komponenten durch Tiefen- oder Breitensuche ausrechnen.

1. Wähle $v \in V(G)$ beliebig.
2. Berechne einen DFS-Baum T_v mit Wurzel v.
   Dann ist $G[V(T_v)]$ eine Komponente von G.
3. Wenn es noch unmarkierte Knoten gibt, wähle einen solchen Knoten v und wiederhole Schritt 2.

Beispiel.

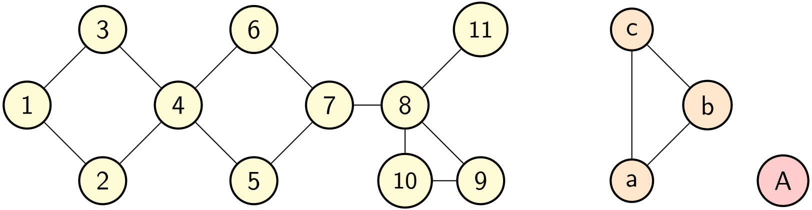


Zusammenhangskomponenten in Graphen

Berechnen von Komponenten. Wir können Zshg.-Komponenten durch Tiefen- oder Breitensuche ausrechnen.

1. Wähle $v \in V(G)$ beliebig.
2. Berechne einen DFS-Baum T_v mit Wurzel v.
   Dann ist $G[V(T_v)]$ eine Komponente von G.
3. Wenn es noch unmarkierte Knoten gibt, wähle einen solchen Knoten v und wiederhole Schritt 2.

Beispiel.



Zusammenhangskomponenten in Graphen

Erinnerung. Sei G ein nicht-leerer Graph.

Eine *Zusammenhangskomponente* von G ist ein maximaler (in Bezug auf die Untergraphenrelation), zusammenhängender Untergraph $C \subseteq G$.

Notation. Statt Zusammenhangskomponente sagt man oft auch nur Komponente.

Beobachtung. Jeder Graph ist die disjunkte Vereinigung seiner Zusammenhangskomponenten.

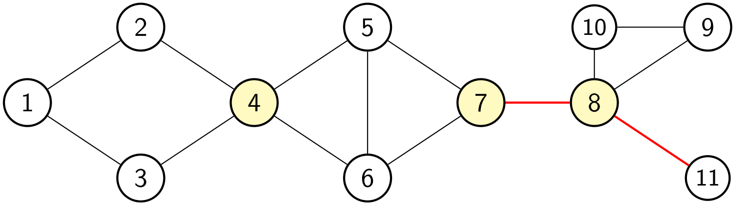
5.9 Mehrfachzusammenhang

Mehrfachzusammenhang

Einfacher Zusammenhang.

Einfacher Zusammenhang bedeutet, dass Informationen etc. von jedem Knoten zu jedem anderen geschickt werden können.

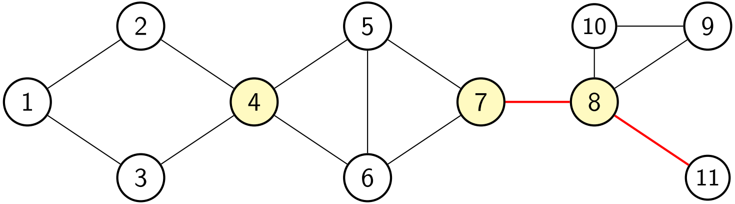
Aber möglicherweise reicht schon der Ausfall eines einzigen Knotens um den Zusammenhang zu zerstören.



Brücken und Schnittknoten

Definition. Sei G ein zusammenhängender Graph.

1. $v \in V(G)$ ist ein *Schnittknoten*, wenn $G - v$ nicht zusammenhängend ist.
2. $e = \{u, v\}$ ist eine *Brücke* von G, wenn $G - e$ nicht zusammenhängend ist.



Brücken und Schnittknoten

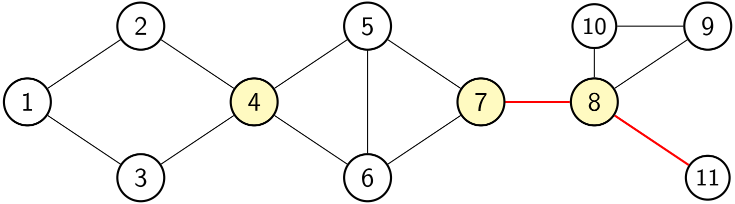
Definition. Sei G ein zusammenhängender Graph.

1. $v \in V(G)$ ist ein *Schnittknoten*, wenn $G - v$ nicht zusammenhängend ist.
2. $e = \{u, v\}$ ist eine *Brücke* von G, wenn $G - e$ nicht zusammenhängend ist.

Definition. Sei G ein beliebiger Graph.

Ein Schnittknoten von G ist ein Schnittknoten einer Komponente von G.

Eine Brücke von G ist eine Brücke einer Komponente von G.



Brücken und Schnittknoten

Fragen.

1. Enthält jeder Graph G, der eine Brücke enthält, auch einen Schnittknoten?

Definition. Sei G ein zusammenhängender Graph.

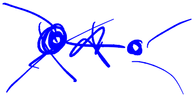
1. $v \in V(G)$ ist ein *Schnittknoten*, wenn $G - v$ nicht zusammenhängend ist.
2. $e = \{u, v\}$ ist eine *Brücke* von G, wenn $G - e$ nicht zusammenhängend ist.

Definition. Sei G ein beliebiger Graph.

Ein Schnittknoten von G ist ein Schnittknoten einer Komponente von G.

Eine Brücke von G ist eine Brücke einer Komponente von G.

Brücken und Schnittknoten



Fragen.

1. Enthält jeder Graph G, der eine Brücke enthält, auch einen Schnittknoten?

   Ja, es sei denn die Brücke verbindet zwei Knoten vom Grad 1.

2. Enthält jeder Graph, der einen Schnittknoten enthält, auch eine Brücke?

Definition. Sei G ein zusammenhängender Graph.

1. $v \in V(G)$ ist ein *Schnittknoten*, wenn $G - v$ nicht zusammenhängend ist.
2. $e = \{u, v\}$ ist eine *Brücke* von G, wenn $G - e$ nicht zusammenhängend ist.

Definition. Sei G ein beliebiger Graph.

Ein Schnittknoten von G ist ein Schnittknoten einer Komponente von G.

Eine Brücke von G ist eine Brücke einer Komponente von G.

Brücken und Schnittknoten

Fragen.

1. Enthält jeder Graph G, der eine Brücke enthält, auch einen Schnittknoten?

   Ja, es sei denn die Brücke verbindet zwei Knoten vom Grad 1.

2. Enthält jeder Graph, der einen Schnittknoten enthält, auch eine Brücke? Nein.

Definition. Sei G ein zusammenhängender Graph.

1. $v \in V(G)$ ist ein *Schnittknoten*, wenn $G - v$ nicht zusammenhängend ist.
2. $e = \{u, v\}$ ist eine *Brücke* von G, wenn $G - e$ nicht zusammenhängend ist.

Definition. Sei G ein beliebiger Graph.

Ein Schnittknoten von G ist ein Schnittknoten einer Komponente von G.

Eine Brücke von G ist eine Brücke einer Komponente von G.

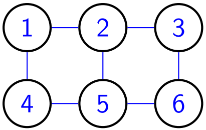
Zusammenhang und k-Zusammenhang

Definition. Sei G ein Graph.

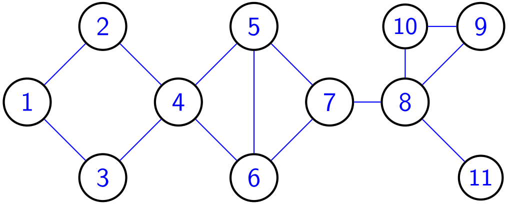
1. Ein Graph G ist *k-zusammenhängend*, wenn $|G| > k$ und $G - X$ für jede Menge $X \subseteq V(G)$ mit $|X| < k$ zusammenhängend ist.
2. Wir bezeichnen das maximale $k \geq 0$ für das G k-zusammenhängend ist als die *Konnektivität*, oder den *Zusammenhang*, $\kappa(G)$ von G.

G

2-Zshgd.



H



1-Zshg.

Zusammenhang vs. 1-Zusammenhang

Erinnerung. *Ein Graph G ist k-zusammenhängend, wenn $|G| > k$ und $G - X$ für jede Menge $X \subseteq V(G)$ mit $|X| < k$ zusammenhängend ist.*

1-Zshg vs Zusammenhang. Zusammenhängende Graphen sind also 1-zusammenhängend oder isolierte Knoten.

Bemerkung. Jeder Graph kann aus 1-zusammenhängenden Graphen und isolierten Knoten durch disjunkte Vereinigung konstruiert werden.

Anders gesagt, kann jeder Graph eindeutig in seine Komponenten zerlegt werden.

Zusammenhang vs. 1-Zusammenhang

Erinnerung. *Ein Graph G ist k-zusammenhängend, wenn $|G| > k$ und $G - X$ für jede Menge $X \subseteq V(G)$ mit $|X| < k$ zusammenhängend ist.*

1-Zshg vs Zusammenhang. Zusammenhängende Graphen sind also 1-zusammenhängend oder isolierte Knoten.

Bemerkung. Jeder Graph kann aus 1-zusammenhängenden Graphen und isolierten Knoten durch disjunkte Vereinigung konstruiert werden.

Anders gesagt, kann jeder Graph eindeutig in seine Komponenten zerlegt werden.

Frage. Können wir die 1-zusammenhängenden Komponenten weiter in ihre 2-zusammenhängenden Teile zerlegen?

Zerlegung in 2-zusammenhängende Teile

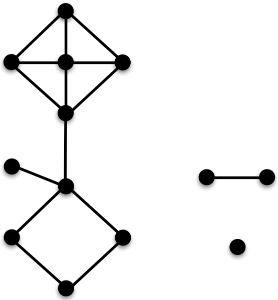
Frage. Können wir die 1-zusammenhängenden Komponenten weiter in ihre 2-zusammenhängenden Teile zerlegen?

Blockgraph. Die andere Sichtweise des Zerlegens 2-fach zusammenhängender Graphen ergibt genau die *Blockgraphen*.

Beobachtung. Wenn G zusammenhängend, aber nicht 2-fach zusammenhängend ist, so hat G einen Schnittknoten oder besteht nur aus einer einzelnen Kante oder einem einzelnen Knoten.

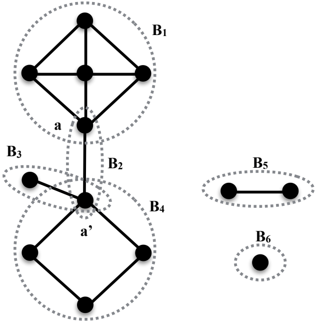
Der Blockgraph

Definition. Sei G ein Graph. Ein maximaler zusammenhängender Untergraph von G ohne Schnittknoten heißt Block von G.



Der Blockgraph

Definition. Sei G ein Graph. Ein maximaler zusammenhängender Untergraph von G ohne Schnittknoten heißt Block von G.



Der Blockgraph

Definition. Sei G ein Graph. Ein maximaler zusammenhängender Untergraph von G ohne Schnittknoten heißt Block von G.

Definition. Sei $A \subseteq V(G)$ die Menge der Schnittknoten in G und sei $\mathcal{B}$ die Menge der Blöcke von G.

二分图

Der Blockgraph von G ist definiert als der bipartite Graph $B(G)$ mit Knotenmenge $A \cup \mathcal{B}$ und Kantenmenge

$$\{\{a, B\} : B \in \mathcal{B}, a \in A \text{ und } a \in V(B)\}.$$

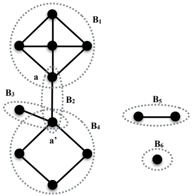
Der Blockgraph

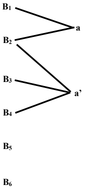
Definition. Sei G ein Graph. Ein maximaler zusammenhängender Untergraph von G ohne Schnittknoten heißt Block von G.

Definition. Sei $A \subseteq V(G)$ die Menge der Schnittknoten in G und sei $\mathcal{B}$ die Menge der Blöcke von G.

Der Blockgraph von G ist definiert als der bipartite Graph $B(G)$ mit Knotenmenge $A \cup \mathcal{B}$ und Kantenmenge

$$\{\{a, B\} : B \in \mathcal{B}, a \in A \text{ und } a \in V(B)\}.$$





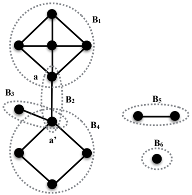
Der Blockgraph

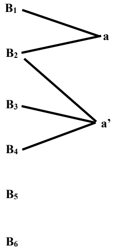
Definition. Sei G ein Graph. Ein maximaler zusammenhängender Untergraph von G ohne Schnittknoten heißt Block von G.

Definition. Sei $A \subseteq V(G)$ die Menge der Schnittknoten in G und sei $\mathcal{B}$ die Menge der Blöcke von G.

Der Blockgraph von G ist definiert als der bipartite Graph $B(G)$ mit Knotenmenge $A \cup \mathcal{B}$ und Kantenmenge

$$\{\{a, B\} : B \in \mathcal{B}, a \in A \text{ und } a \in V(B)\}.$$





Proposition. Für jeden Graph G ist der Blockgraph $B(G)$ ein Wald.

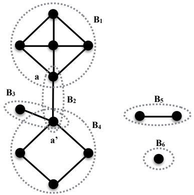
Ist G zusammenhängend, dann ist $B(G)$ ein Baum.

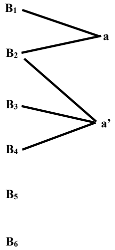
Blockgraphen

Satz. Der Blockgraph eines Graphs G ist azyklisch.

Beweis.

Sei G ein Graph und seien $\mathcal{B}$ die Blöcke und A die Menge der Schnittknoten von G. Angenommen, der Satz wäre falsch.





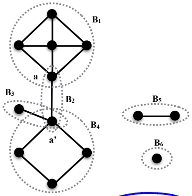
Blockgraphen

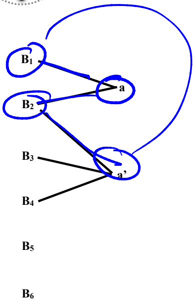
Satz. Der Blockgraph eines Graphs G ist azyklisch.

Beweis.

Sei G ein Graph und seien $\mathcal{B}$ die Blöcke und A die Menge der Schnittknoten von G. Angenommen, der Satz wäre falsch.

Dann gibt es einen Kreis $C := a_1 B_1 a_2 B_2 \dots a_n B_n$ in $B(G)$. Wir definieren $a_{n+1} := a_1$.





Blockgraphen

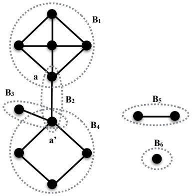
Satz. Der Blockgraph eines Graphs G ist azyklisch.

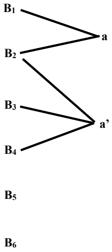
Beweis.

Sei G ein Graph und seien $\mathcal{B}$ die Blöcke und A die Menge der Schnittknoten von G. Angenommen, der Satz wäre falsch.

Dann gibt es einen Kreis $C := a_1 B_1 a_2 B_2 \ldots a_n B_n$ in $B(G)$. Wir definieren $a_{n+1} := a_1$.

Nach Definition gilt $a_i, a_{i+1} \in V(B_i)$ für alle $1 \leq i \leq n$.





Blockgraphen

Satz. Der Blockgraph eines Graphs G ist azyklisch.

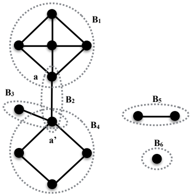
Beweis.

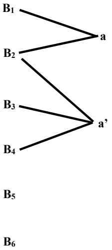
Sei G ein Graph und seien $\mathcal{B}$ die Blöcke und A die Menge der Schnittknoten von G. Angenommen, der Satz wäre falsch.

Dann gibt es einen Kreis $C := a_1 B_1 a_2 B_2 \dots a_n B_n$ in $B(G)$. Wir definieren $a_{n+1} := a_1$.

Nach Definition gilt $a_i, a_{i+1} \in V(B_i)$ für alle $1 \leq i \leq n$.

Da alle $B \in \mathcal{B}$ zusammenhängend sind, existiert in B_i also ein Pfad P_i von a_i nach a_{i+1}.





Blockgraphen

Satz. Der Blockgraph eines Graphs G ist azyklisch.

Beweis.

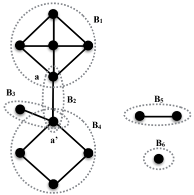
Sei G ein Graph und seien $\mathcal{B}$ die Blöcke und A die Menge der Schnittknoten von G. Angenommen, der Satz wäre falsch.

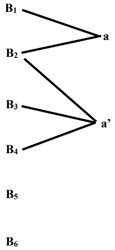
Dann gibt es einen Kreis $C := a_1 B_1 a_2 B_2 \dots a_n B_n$ in $B(G)$. Wir definieren $a_{n+1} := a_1$.

Nach Definition gilt $a_i, a_{i+1} \in V(B_i)$ für alle $1 \leq i \leq n$.

Da alle $B \in \mathcal{B}$ zusammenhängend sind, existiert in B_i also ein Pfad P_i von a_i nach a_{i+1}.

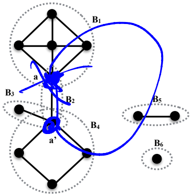
Da in C kein Schnittknoten und kein Block wiederholt wird, liefert $P_1 \cup P_2 \dots P_n$ einen Kreis in G der mindestens zwei Schnittknoten und mindestens zwei Blöcke enthält.





Blockgraphen

Satz. Der Blockgraph eines Graphs G ist azyklisch.



Beweis.

Sei G ein Graph und seien $\mathcal{B}$ die Blöcke und A die Menge der Schnittknoten von G. Angenommen, der Satz wäre falsch.

Dann gibt es einen Kreis $C := a_1 B_1 a_2 B_2 \dots a_n B_n$ in $B(G)$. Wir definieren $a_{n+1} := a_1$.

Nach Definition gilt $a_i, a_{i+1} \in V(B_i)$ für alle $1 \leq i \leq n$.

Da alle $B \in \mathcal{B}$ zusammenhängend sind, existiert in B_i also ein Pfad P_i von a_i nach a_{i+1}.

Da in C kein Schnittknoten und kein Block wiederholt wird, liefert $P_1 \cup P_2 \dots P_n$ einen Kreis in G der mindestens zwei Schnittknoten und mindestens zwei Blöcke enthält.

Dann sind aber B_1 und B_2 keine Blöcke, denn durch Löschen eines einzigen Knotens kann man B_1 und B_2 nicht trennen. $\square$