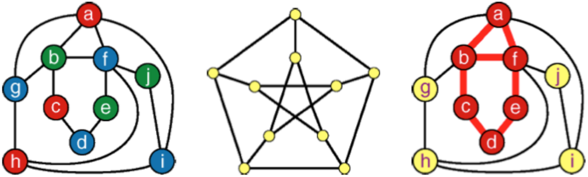
Algorithmische Graphentheorie

Vorlesung 2: Einführung in die Graphentheorie - Teil 2

Babeş-Bolyai Universität, Department für Informatik, Cluj-Napoca
csacarea@cs.ubbcluj.ro



Definition

Seien $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$ zwei Graphen mit disjunkten Knotenmengen ($V_1 \cap V_2 = \emptyset$). Die disjunkte Vereinigung der beiden Graphen ist der Graph

$$G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2).$$

Operationen mit Graphen

Vereinigung

Falls $V_1 = V_2$, die Vereinigung $G_1 \cup G_2$ ist definiert als der Graph mit allen Kanten aus G_1 und G_2.

Im Allgemeinen, ist die Vereinigung zweier Graphen $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$ definiert als

$$G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2),$$

mit $V_1 \subseteq V_2$ oder $V_2 \subseteq V_1$ oder $V_1 = V_2$ oder $V_1 \cap V_2 = \emptyset$.

Definition

Seien $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$ zwei Graphen. Der ***Durchschnitt*** *der beiden Graphen ist der Graph*

$$G_1 \cap G_2 = (V_1 \cap V_2, E_1 \cap E_2).$$

Beispiel

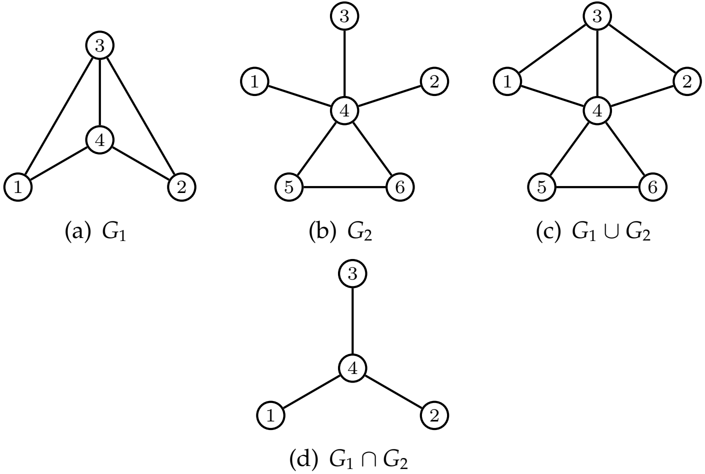


Abbildung 1: Vereinigung und Durchschnitt von Graphen.

Definition

Die symmetrische Differenz zweier Graphen $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$ ist definiert als der Graph

$$G_1 \Delta G - 2 = (V, E),$$

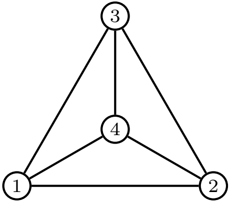
mit $V = V_1 \Delta V_2$ und die Kantenmenge ist definiert als

$$E = (E_1 \Delta E_2) \setminus \{uv \mid u \in V_1 \cap V_2 \quad or \quad v \in V_1 \cap V_2\}.$$

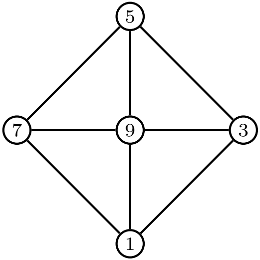
Zur Erinnerung, die symmetrische Differenz zweier Mengen S_1 und S_2 ist definiert als

$$S_1 \Delta S_2 = \{x \in S_1 \cup S_2 \mid x \notin S_1 \cap S_2\}.$$

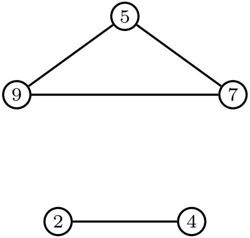
BEISPIEL



(a) G_1



(b) G_2

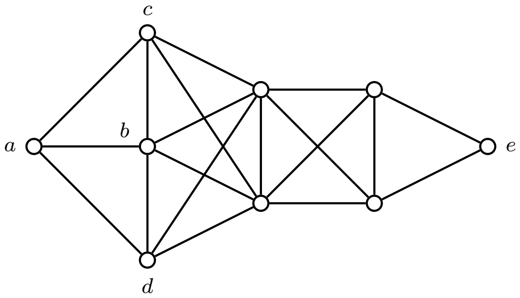


(c) $G_1 \Delta G_2$

Entfernen von Knoten und Kanten

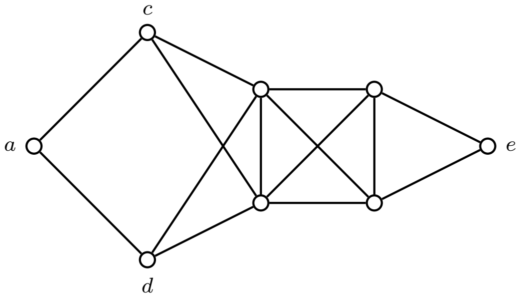
- Sei $G = (V, E, \gamma)$ ein Graph. Das Entfernen einer Kante $e \in E$ erzeugt aus G einen neuen Graphen $G - \{e\} = (V, E \setminus \{e\}, \gamma)$.
- Analog für eine Kantenmenge $F \subseteq E$.
- $G - \{v\}$ der Graph, der aus G durch Entfernen des Knotens v hervorgeht.
- Das Entfernen von v schließt das gleichzeitige Entfernen aller zu v inzidierenden Kanten ein.
- Analog für einen Kantenmenge $X \subseteq V$

Beispiel



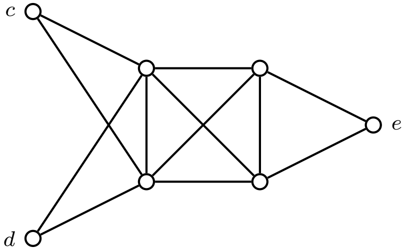
(a) G

Beispiel



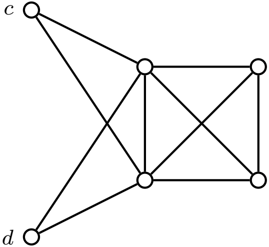
(b) $G - \{b\}$

Beispiel



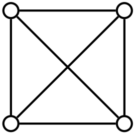
(c) $G - \{a, b\}$

Beispiel



(d) $G - \{a, b, e\}$

Beispiel



(e) $G - \{a, b, c, d, e\}$

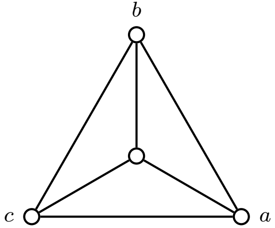
Beispiel mit SAGE

Knotenentfernung

```
sage: G = Graph({1:[2,4], 2:[1,4], 3:[2,6], 4:[1,3], 5:[4,2], 6:[3,1]})
sage: G.vertices()
[1, 2, 3, 4, 5, 6]
sage: E1 = Set(G.edges(labels=False)); E1
{(1, 2), (4, 5), (1, 4), (2, 3), (3, 6), (1, 6), (2, 5), (3, 4), (2, 4)}
sage: E4 = Set(G.edges_incident(vertices=[4], labels=False)); E4
{(4, 5), (3, 4), (2, 4), (1, 4)}
sage: G.delete_vertex(4)
sage: G.vertices()
[1, 2, 3, 5, 6]
sage: E2 = Set(G.edges(labels=False)); E2
{(1, 2), (1, 6), (2, 5), (2, 3), (3, 6)}
sage: E1.difference(E2) == E4
True
```

Beispiel

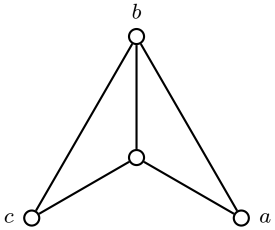
Kantenentfernung



(f) G

Beispiel

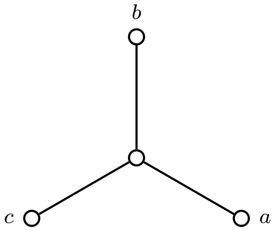
Kantenentfernung



(g) $G - \{ac\}$

Beispiel

Kantenentfernung



(h) $G - \{ab, ac, bc\}$

Beispiel mit SAGE

Kantenentfernung

```
sage: V1 = G.vertices(); V1
[1, 2, 3, 4, 5, 6]
sage: E14 = Set([(1,4)]); E14
{(1, 4)}
sage: G.delete_edge([1,4])
sage: E2 = Set(G.edges(labels=False)); E2
{(1, 2), (4, 5), (2, 3), (3, 6), (1, 6), (2, 5), (3, 4), (2, 4)}
sage: E1.difference(E2) == E14
True
```

Fusion und Kontraktion

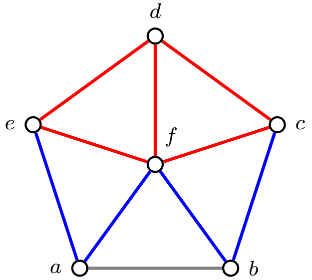
Fusion

Identifizieren der Knoten v und w in einem Knoten, der zu allen Kanten inzident ist, die vorher einen dieser Knoten als Endknoten hatten. Wir bezeichnen den entstehenden Graphen mit G_{uv}. (Analog G_X).

Kontraktion

...der Kante e, mit $\gamma(e) = \{u, v\}$ ist das Entfernen von e mit der anschließenden Fusion der Endknoten u und v. Wir bezeichnen den durch Kontraktion von e aus G hervorgehenden Graphen mit G/e.

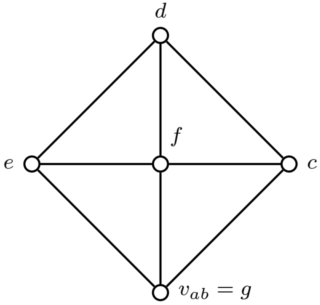
Beispiel
Kontraktion



(i) G_1

Beispiel

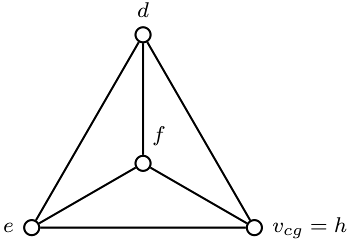
Kontraktion



(j) $G_2 = G_1/ab$

Beispiel

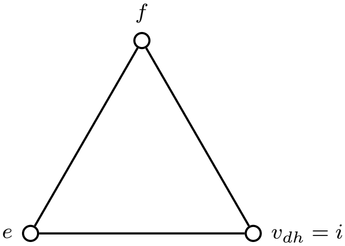
Kontraktion



(k) $G_3 = G_2/cg$

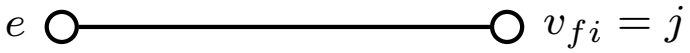
Beispiel

Kontraktion



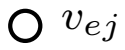
(l) $G_4 = G_3/dh$

Beispiel
Kontraktion



(m) $G_5 = G_4/fi$

BEISPIEL
KONTRAKTION

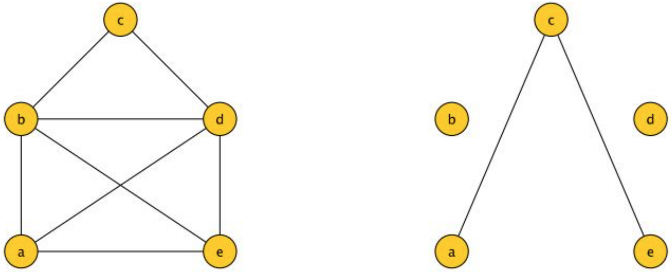


(n) $G_6 = G_5/ej$

Komplementgraph

Definition

Ein Komplementgraph ist ein Graph mit gleicher Knotenmenge aber die Kantenmenge besteht aus genau die Knoten, die im Ursprungsgraph nicht vorhanden sind.
Ein schlichter Graph, der isomorph ist zu seinem Komplementgraph, heißt Selbstkomplementär.



Theorem

Falls der Graph $G = (V, E)$ *Selbstkomplementär ist, dann ist die Ordnung von G gleich mit* $|V| = 4k$ *oderr* $|V| = 4k + 1$ *für* $k \in \mathbb{N}$. *Falls die Ordnung von G gleich* $n = |V|$ *ist, dann gilt* $|E| = n(n-1)/4$.

Kartesischer Produkt

Definition

Der *kartesischer Produkt* $G \square H$ *der Graphen* G *und* H *ist ein Graph so dass die Menge der Knoten von* $G \square H$ *ist das kartesische Produkt*

$$V(G \square H) = V(G) \times V(H).$$

Zwei Kanten (u, u') *und* (v, v') *sind adjazent in* $G \square H$ *genau dann, wenn entweder*

1. $u = v$ *und* u' *ist adjazent zu* v' *in* H*; oder*
2. $u' = v'$ *und* u *ist adjacent zu* v *in* G.

Die Knotenmenge von $G \square H$ *ist* $V(G \square H)$ *und die Kantenmenge von* $G \square H$ *ist*

$$E(G \square H) = \big(V(G) \times E(H)\big) \cup \big(E(G) \times V(H)\big).$$

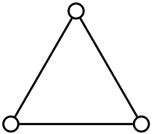
Beispiel mit SAGE

Kartesisches Produkt

```
sage: Z = graphs.CompleteGraph(2); len(Z.vertices()); len(Z.edges())
2
1
sage: C = graphs.CycleGraph(5); len(C.vertices()); len(C.edges())
5
5
sage: P = C.cartesian_product(Z); len(P.vertices()); len(P.edges())
10
15
```

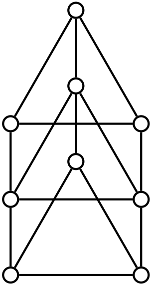
Beispiel mit SAGE

Kartesisches Produkt



(o) K_3

(p) P_3



(q) $K_3 \square P_3$

n-DIMENSIONALER HYPERCUBE

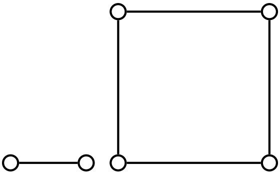
Definition

Der *n-dimensionale Hypercube* $Q_n = (V_n, E_n)$ *ist definiert wie folgt:*

- V_n *ist die Menge der Bitstrings der Länge* n.
- *Für zwei Bitstrings* $p, q \in V_n$ *gilt* $\{p, q\} \in E_n$ *genau dann, wenn* p *und* q *sich in genau einem Bit unterscheiden.*

Das kartesische Produkt von K_2 Graphen ist das Hypercube:

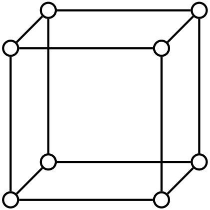
$$(K_2)^{\Box n} = Q_n.$$



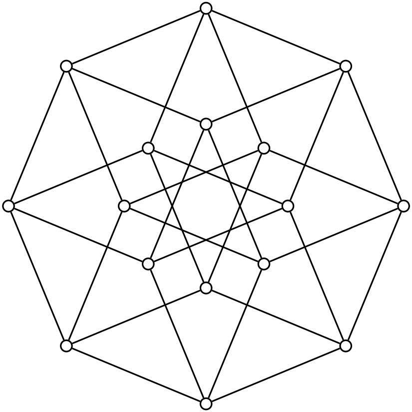
(r) Q_1 (s) Q_2

Beispiel

Hypercube Graphen



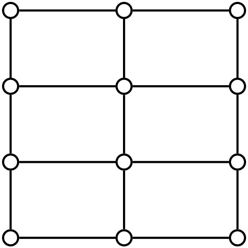
(t) Q_3



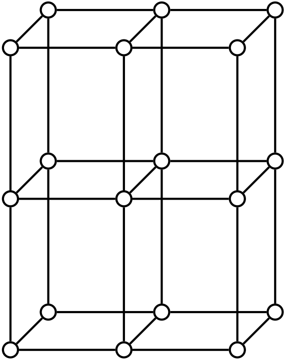
(u) Q_4

Beispiel

Meshgraphen



(v) $M(3,4)$



(w) $M(3,2,3)$

Minoren

Definition

Ein Graph H heißt Minor *eines Graphen G, falls H isomorph ist zu einem Graph, welcher als eine Reihenfolge von Kantenkontraktionen aus G entsteht.*

Untergraph

Definition

Sei $G = (V, E)$ ein Graph. Ein Graph $H = (W, F)$ mit $W \subseteq V$ und $F \subseteq E$ heißt Untergraph (subgraph) von G.

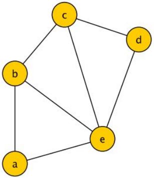
Gilt $W = V$, dann heißt H aufspannender Untergraph (spanning subgraph) von G.

Gilt

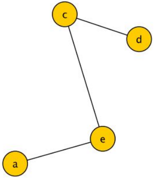
$$F = \{\{v, w\} \mid \{v, w\} \in E, v, w \in W\},$$

dann heißt H induzierter Untergraph (induced subgraph) von G. Für solch einen induzierten Untergraphen schreiben wir auch $G(W)$.

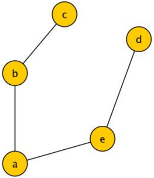
Untergraph (2)



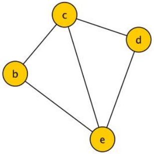
G



Untergraph
nicht aufspannend
nicht induziert

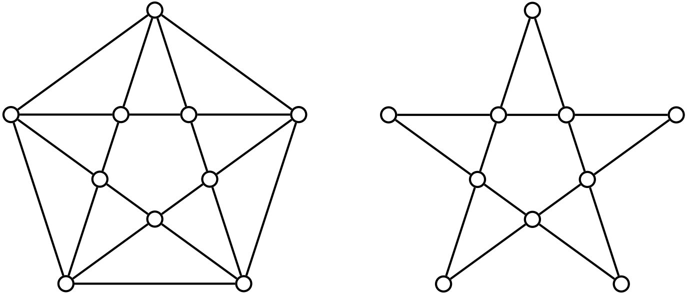


aufspannender Untergraph



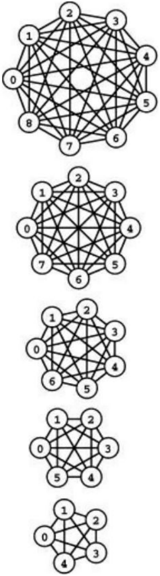
induzierter Untergraph

UNTERGRAPH (3)



(x) (y)

UNTERGRAPH (4)



Untergraph (5)

- Jeder Graph mit mindestens 5 und höchstens 9 Knoten ist ein Untergraph dieser vollständiger Graphen.
- Wieviele gibt es?

UNTERGRAPH (5)

- Jeder Graph mit mindestens 5 und höchstens 9 Knoten ist ein Untergraph dieser vollständiger Graphen.
- Wieviele gibt es? $\geq$ 68 Milliarden...

ANZAHL DER KANTEN

Sei $G = (V, E, \gamma)$ ein endlicher Graph mit $|V| = n$. Wie groß ist die maximale Anzahl der möglichen Kanten?

Anzahl der Kanten

Sei $G = (V, E, \gamma)$ ein endlicher Graph mit $|V| = n$. Wie groß ist die maximale Anzahl der möglichen Kanten?

Satz

Ein Graph mit n Knoten hat maximal $n(n-1)/2$ Kanten

Beweis:

- Es gibt höchstens n^2 mögliche Kanten,
- davon n Schlingen,
- Alle Kanten sind doppelt gezählt.
- Somit beträgt die maximale Anzahl der Kanten $(n^2 - n)/2 = n(n-1)/2$

Clique

Definition

Es sei $G = (V, E)$ ein Graph. Eine Knotenmenge $U \subseteq V$ (bzw. der von U induzierte Untergraph $G(U)$) heißt Clique genau dann, wenn $G(U)$ ein vollständiger Graph ist.

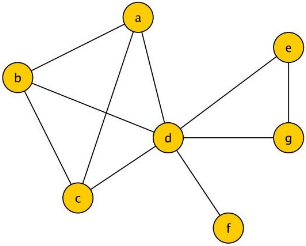
Die maximale Größe einer Clique in G wird mit $\omega(G)$ bezeichnet, d.h.

$$\omega(G) := \max\{|U| \mid U \text{ ist Clique in } G\}.$$

CLIQUE (2)

Beispiel

- $\{a, b, c, d\}$ bildet eine Clique der Größe 4.
- $\{d, e, g\}$ bildet eine Clique der Größe 3.
- $\{d, f\}$ bildet eine Clique der Größe 2.
- $\omega(G) = 4$



WEGE

Definition

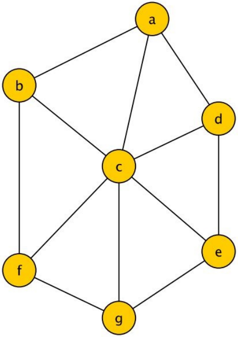
Es sei $G = (V, E)$ ein Graph.

- *Eine Folge $(v_0, v_1, \ldots, v_n)$ von Knoten mit $e_i := \{v_{i-1}, v_i\} \in E$ für $i = 1, 2, \ldots, n$ heißt Kantenzug (walk). Die Anzahl der Kanten in einem Kantenzug ist die Länge dieses Kantenzugs.*
- *Ein Kantezug, bei dem die Kanten e_i alle verschieden sind, heißt Weg (trail). Die Länge des Weges ist n.*
- *Ein Weg heißt einfacher Weg (path) gdw. die Knoten v_j paarweise verschieden sind.*

WEGE (2)

Beispiel

- (a, b, c, a, b, f) ist ein Kantenzug, aber kein Weg.
- (c, b, f, c, d) ist ein Weg, aber kein einfacher Weg.
- (a, b, f, c, d) ist ein einfacher Weg.



Kreise

Definition

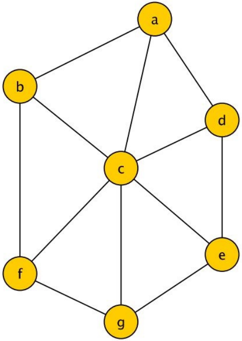
Die folgenden Bezeichnungen beziehen sich auf die vorige Definition:

- *Gilt in einem Kantezug $v_0 = v_n$, so sprechen wir von einem geschlossenen Kantenzug (closed walk).*
- *Ein Weg für den $v_0 = v_n$ gilt heißt Kreis (closed trail).*
- *Ein Kreis, bei dem die Knoten v_j mit Ausnahme von $v_0 = v_n$ paarweise verschieden sind, heißt einfacher Kreis (cycle).*

Kreise (2)

Beispiel

- (a, b, c, a, d, c, a) ist ein geschlossener Kantenzug, aber kein Kreis.
- (b, c, e, d, c, f, b) ist ein Kreis, aber kein einfacher Kreis.
- (a, b, f, c, a) ist ein einfacher Kreis.



Bemerkungen zu Wege und Kreise

- Ein Knoten alleine stellt einen Kreis der Länge 0 dar. Im folgenden ist Kreis immer ein nichttrivialer Kreis gemeint, d.h. ein Kreis mit Länge > 0.
- Nur in schlichten Graphen ist durch die Knotenfolge der Weg bzw. Kreis eindeutig bestimmt.
- In schlichten Graphen existieren keine Kreise der Länge 1 und 2.

Hilfssätze zu Wege und Kreise

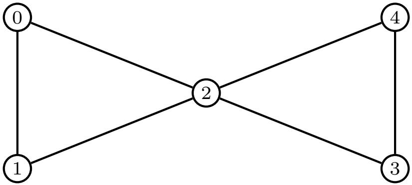
Lemma

Es sei $G = (V, E)$ ein Graph und es seien $a, b \in V, a \neq b$ zwei verschieden Knoten von G. Dann gilt: Wenn ein Kantenzug von a nach b existiert, dann existiert auch ein einfacher Weg von a nach b.

Lemma

Wenn ein Graph G einen geschlossenen Kantenzug K enthält, in dem eine Kante von K nicht mehrfach vorkommt, dann enthält G auch einen einfachen Kreis.

Schmetterlinggraph mit 5 Kanten



Aufgabe

Sei G der Schmetterlinggraph mit 5 Kanten.

1. Finde zwei verschiedene Kantenzüge, welche keine Wege sind und bestimme deren Länge.
2. Bestimme zwei verschiedene Wege, welche keine Pfade sind und bestimme deren Länge.
3. Finde zwei verschiedene Pfade und bestimme deren Länge.
4. Finde einen geschlossenen Weg, der kein Kreis ist.
5. Finde einen geschlossenen Kantenzug C, der eine Kante e besitzt, so dass $C - \{e\}$ einen Kreis enthält.

Theorem

Jeder $u - v$ Kantenzug in einem Graphen enthält ein $u - v$ Pfad.

Beweis

- Ein Kantenzug der Länge $n = 0$ ist ein trivialer Pfad.
- Sei W ein $u - v$ Kantenzug der Länge $n > 0$ in einem Graph G:

  $$W : u = v_0, v_1, \dots, v_n = v.$$

  Es ist möglich, dass ein Knoten aus W sich wiederholt. Falls dies nicht der Fall ist, dann ist W ein Pfad. Seien $0 \leq i, j \leq n$ mit $i < j$ und $v_i = v_j$. Durch Entfernen der Knoten $v_i, v_{i+1}, \dots, v_{j-1}$ aus W entsteht ein $u - v$ Kantenzug W_1 der Länge kleiner als n. Falls W_1 ein Pfad ist, dann sind wir fertig. Falls nicht, wiederholen wir diese Prozedur. Wegen der Endlichkeit von W erhalten wir ein Pfad nach endlich vielen Schritten.