Graphentheorie

Grundlagen

Andreas Hohenauer

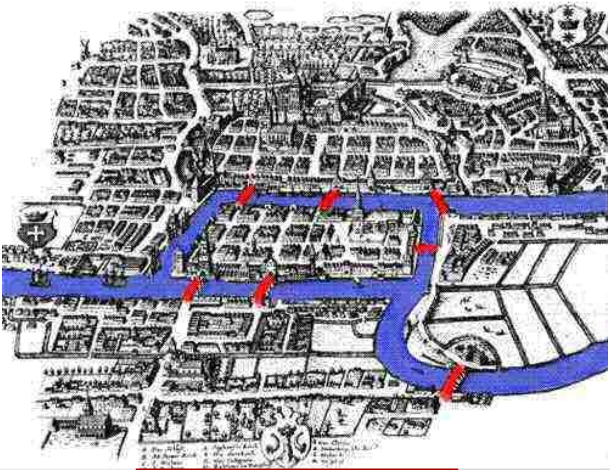
Algorithmen und Datenstrukturen

12. September 2025

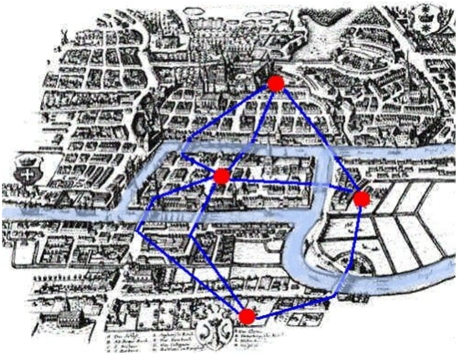
Anfänge der Graphentheorie 1736:

Leonhard Euler untersucht das *“Königsberger Brückenproblem”*

Fragestellung: Gibt es einen Weg der jede der sieben Brücken über den Fluss Pregel *genau einmal* benutzt?

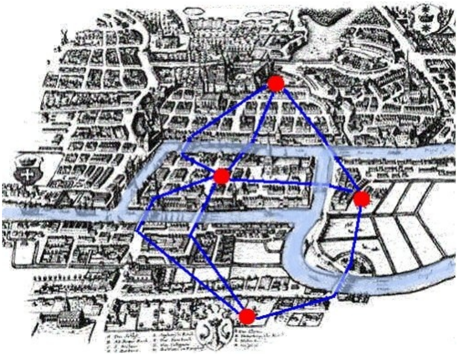


Euler modellierte erstmals eine Problemstellung mit einem **Graphen**



Die roten Kreise werden als *Knoten* bezeichnet, die blauen Verbindungen als *Kanten*.

Euler modellierte erstmals eine Problemstellung mit einem **Graphen**

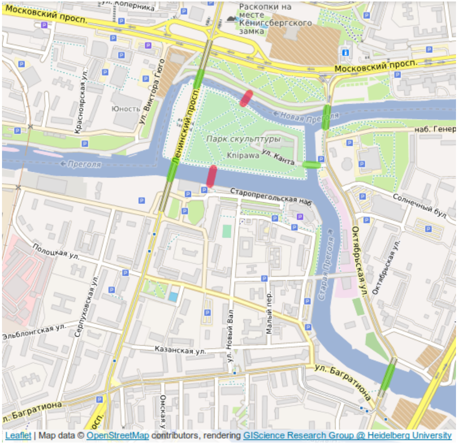


Die roten Kreise werden als *Knoten* bezeichnet, die blauen Verbindungen als *Kanten*.

Euler fand heraus: so ein Rundgang existiert nicht!

Seit 1946 trägt Königsberg den Namen *Kaliningrad* und ist eine russische Exklave nördlich von Polen.

Von den historischen sieben Brücken existieren heute nur noch fünf!

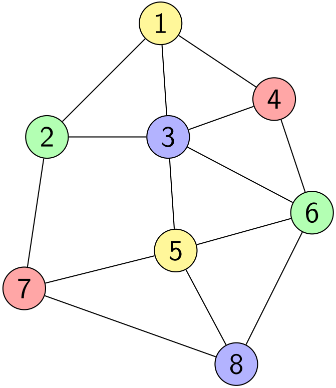


Navigationssysteme

- Berechnung von kürzesten Wegen in Navigationssystemen
- Straßennetz wird als Graph repräsentiert

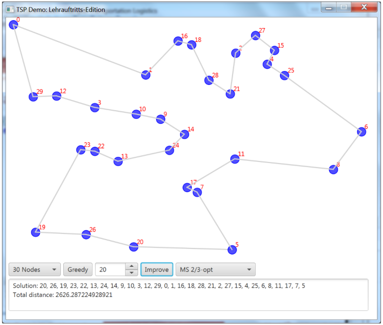
Mobilfunk

- Frequenzplanung in der Mobilfunkindustrie
- Interferenzen bei gleichen Frequenzen und nahen Senderstandorten
- Modellierung durch Graphen:
  - Ordne jedem Knoten eine Frequenz ("Farbe") zu
  - Benachbarte Knoten dürfen nicht die selbe Farbe haben
  - Finde minimale Anzahl an Farben: Optimierungsproblem



Problem des Handlungsreisenden

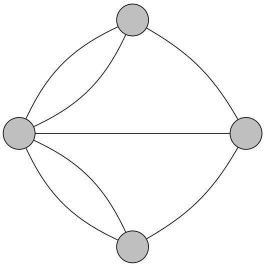
- "Travelling Salesman Problem" (TSP)
- Archetypisches Optimierungsproblem auf Graphen
- Problemstellung: Handlungsreisender möchte bestimmte Orte jeweils einmal besuchen
- Gesucht: Kürzester Weg
- Extrem schwierig (optimal) zu lösen.
- Fundamentale Bedeutung für viele Anwendungen in Logistik, Verbindungen auf Leiterplatten



Weitere Einsatzgebiete

- Theoretische Informatik, formale Sprachen, Compilerbau
- Datenstrukturen (beim Programmieren)
- Analyse von Abhähgigkeiten, Prozessplanung, Betriebssysteme
- Standortplanung
- Kommunikationsnetzwerke
- Schneeräumung, Postzustellung, Krankentransport
- Netzwerkdesign: Telekommunikation, Energie, Luftverkehr
- Biomedizin, Bioinformatik
- Produktionsplanung
- Modellierung von Teilproblemen in zahlreichen weiteren Aufgabestellungen

Graphen



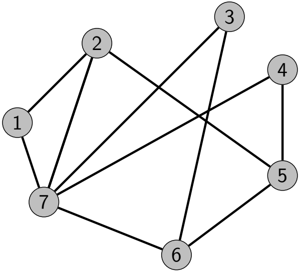
- Die Abbildung zeigt nochmals den Graphen des Königsberger Brückenproblems
- Die grauen Kreise werden **Knoten** *(engl. vertex, vertices [pl.])* genannt
- Die Verbindungen zwischen den Knoten werden **Kanten** *(engl. edges)* genannt
- Das Brückenproblem hat keine Lösung, weil von den Knoten eine ungerade Anzahl von Kanten weggeht.
- Damit ein derartiger *Eulerscher Weg* existiert, dürfen jedoch maximal zwei Knoten ungeraden *Grad* aufweisen.

Ungerichtete Graphen

Definition (Ungerichteter Graph)

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

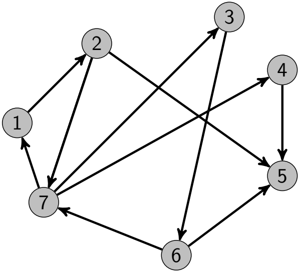
- einer (endlichen) Menge an *Knoten* $V(G) = \{1, 2, \ldots, n\}$; die Knoten werden meist mit Großbuchstaben oder Zahlen bezeichnet.
- einer (endlichen) Menge an *Kanten* $E(G)$; Eine Kante zwischen i and j wird mit $[i, j]$ oder $\{i, j\}$ bezeichnet.



Anmerkung: Die Bezeichnung V für die Knotenmenge stammt von der englischen Bezeichnung *"vertices"* (pl.), bzw. *"vertex"* (sing.), was wörtlich übersetzt *Eckpunkt* bedeutet. Dennoch ist die Bezeichnung *Knoten* auf Deutsch gebräuchlicher. Die Kantenmenge E stammt vom englischen Begriff *"edge(s)"*.

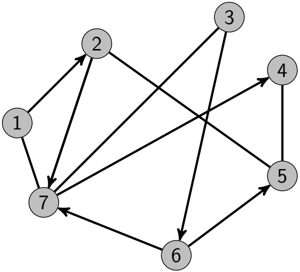
Gerichtete Graphen

- Ungerichtete Kanten haben wir in eckigen Klammern notiert (z.B. $[1, 3]$): Reihenfolge der Knoten spielt keine Rolle!
- Ein *gerichteter Graph* enthält nur *gerichtete Kanten*
- **Def. gerichtete Kante:** (i, j) ist eine gerichtete Verbindung vom Knoten i zum Knoten j
- Eine gerichtete Kanten wird auch *Bogen* oder *Pfeil* genannt. (engl.: *arc*)



Gemischte Graphen

- *Gemischte Graphen* enthalten sowohl gerichtete als auch ungerichtete Kanten



Adjazenz

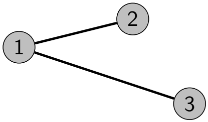
Definition (Adjazente Knoten)

Zwei Knoten i und j sind adjazent, wenn eine Kante $[i, j]$ existiert.

Definition (Adjazente Kanten)

Zwei Kanten $[i, j]$ und $[j, k]$ sind adjazent wenn sie einen gemeinsamen Knoten j besitzen.

Beispiel: im folgenden Graph sind die Kanten $[1, 2]$ und $[1, 3]$ adjazent. Ebenso sind die Knoten 1 und 2, sowie die Knoten 1 und 3 adjazent.

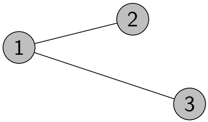


Inzidenz

Definition (Inzidenz)

Ein Knoten ist *inzident* mit einer Kante wenn der Knoten Endpunkt dieser Kante ist.

Beispiel:



Hier ist die Kante $[1,2]$ und $[1,3]$ jeweils inzident mit dem Knoten 1.

Knotengrad

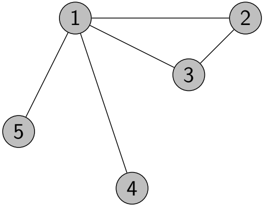
Definition (Knotengrad)

Unter dem Grad (engl. degree) $d(v)$ eines Knotens $v \in V(G)$ versteht man die Anzahl der Kanten $e \in E(G)$ die mit v verbunden sind.

Definition (Endpunkt)

Ist $d(v) = 1$ nennt man v einen *Endpunkt* des Graphen.

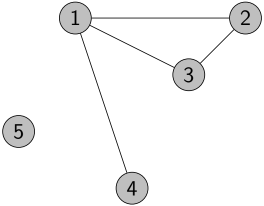
Beispiel: $d(1) = 4, d(2) = 2, d(3) = 2, d(4) = 1, d(5) = 1;$



Isolierter Knoten

Einen Knoten v mit $d(v) = 0$ nennt man *isolierten Knoten*.

Beispiel: $d(5) = 0$

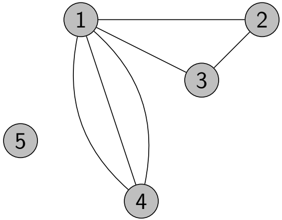


Mehrfachkante

Definition (Mehrfachkante)

Verlaufen zwischen zwei Knoten mindestens zwei Kanten, so heißt diese Menge von Kanten *Mehrfachkante* oder *Multikante*.

Beispiel: Mehrfachkante zwischen Knoten 1 und 4

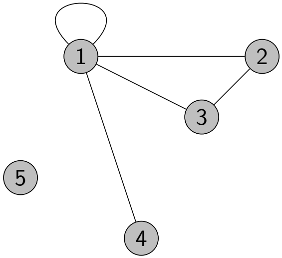


Schlinge

Definition (Schlinge)

Als *Schlinge* wird in einem Graphen eine Kante bezeichnet, die einen Knoten mit sich selbst verbindet.

Beispiel: Schlinge $[1, 1]$



Schlichter Graph

Definition (Schlichter Graph)

Unter einem *schlichten Graphen* (auch: einfacher Graph) versteht man einen Graphen ohne Schlingen und Mehrfachkanten.

Anmerkung: Wir betrachten, sofern nicht anders angegeben, immer schlichte Graphen!

Beispiel

Anmerkung: Ein Graph G ist ausschließlich durch die *Menge* V der Knoten und die *Menge* E der Kanten bestimmt. Wie genau die Knoten und Kanten aufgezeichnet werden, ist nicht relevant!

Beispiel: Graph $G = (V, E)$

Beispiel

Anmerkung: Ein Graph G ist ausschließlich durch die *Menge* V der Knoten und die *Menge* E der Kanten bestimmt. Wie genau die Knoten und Kanten aufgezeichnet werden, ist nicht relevant!

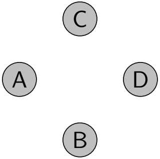
Beispiel: Graph $G = (V, E)$
mit $V = \{A, B, C, D\}$

Beispiel

Anmerkung: Ein Graph G ist ausschließlich durch die *Menge* V der Knoten und die *Menge* E der Kanten bestimmt. Wie genau die Knoten und Kanten aufgezeichnet werden, ist nicht relevant!

Beispiel: Graph $G = (V, E)$
mit $V = \{A, B, C, D\}$

Eine mögliche Darstellung des Graphen sieht wie folgt aus:

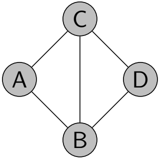


Beispiel

Anmerkung: Ein Graph G ist ausschließlich durch die *Menge* V der Knoten und die *Menge* E der Kanten bestimmt. Wie genau die Knoten und Kanten aufgezeichnet werden, ist nicht relevant!

Beispiel: Graph $G = (V, E)$
mit $V = \{A, B, C, D\}$
und $E = \{[A, B], [A, C], [B, C], [B, D], [C, D]\}$

Eine mögliche Darstellung des Graphen sieht wie folgt aus:

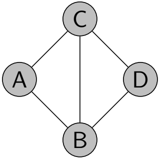


Beispiel

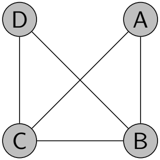
Anmerkung: Ein Graph G ist ausschließlich durch die *Menge* V der Knoten und die *Menge* E der Kanten bestimmt. Wie genau die Knoten und Kanten aufgezeichnet werden, ist nicht relevant!

Beispiel: Graph $G = (V, E)$
mit $V = \{A, B, C, D\}$
und $E = \{[A, B], [A, C], [B, C], [B, D], [C, D]\}$

Eine mögliche Darstellung des Graphen sieht wie folgt aus:



Eine andere Darstellung ist:



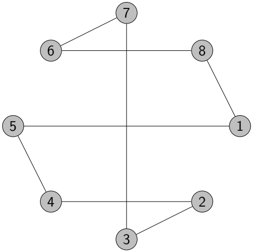
k-regulärer Graph (1)

Definition (Regulärer Graph)

Ein einfacher ungerichteter Graph G heißt *regulär* vom Grad k wenn alle Knoten v den gleichen Knotengrad k haben, also:

$$\forall v \in V : d(v) = k.$$

Beispiel: 2-regulärer Graph



Handshaking Lemma

Eine Kante trägt zur Summe aller Knotengrade den Wert 2 bei. Somit muss die Summe aller Knotengrade immer gerade sein. Insbesondere gilt für **alle ungerichteten Graphen** das

Lemma (Handshaking-Lemma)

$$\sum_{v \in V} d(v) = 2 \cdot |E|$$

Folgerung: Jeder ungerichtete Graph besitzt eine *gerade Anzahl* an Knoten mit *ungeradem Grad*.

Beispiel: 4-regulärer Graph mit 12 Knoten

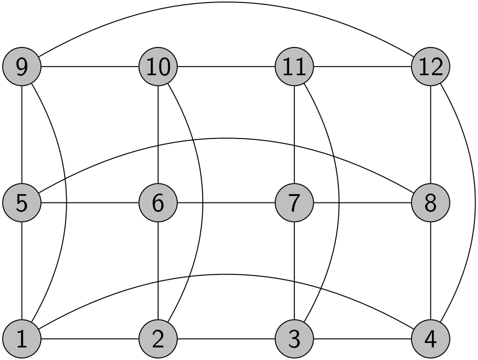
Übungsaufgabe

Konstruieren Sie einen zusammenhängenden, schlichten 4-regulären Graphen mit 12 Knoten

Beispiel: 4-regulärer Graph mit 12 Knoten

Übungsaufgabe

Konstruieren Sie einen zusammenhängenden, schlichten 4-regulären Graphen mit 12 Knoten



Reguläre Graphen

Aufgabe: 3-regulärer Graph

Konstruieren Sie einen schlichten und zusammenhängenden 3-regulären Graphen mit 5 Knoten.

Reguläre Graphen

Aufgabe: 3-regulärer Graph

Konstruieren Sie einen schlichten und zusammenhängenden 3-regulären Graphen mit 5 Knoten.

Dies ist aufgrund des Handshaking-Lemmas unmöglich. Die Summe aller Knotengrade $\sum_{v \in V} d(v) = 5 \cdot 3 = 15$. Somit müsste gelten $|E| = {}^{15}/_{2}$, was nicht möglich ist, da die Anzahl der Kanten stets ganzzahlig sein muss!

Vollständiger Graph (1)

Definition (Vollständiger Graph K_n)

Ein Graph G mit $n = |V|$ Knoten heißt vollständiger Graph K_n, wenn er regulär vom Grad $n - 1$ ist.

Bemerkung: In einem vollständigen Graphen sind somit alle Knoten direkt miteinander verbunden.

Vollständiger Graph (1)

Definition (Vollständiger Graph K_n)

Ein Graph G mit $n = |V|$ Knoten heißt vollständiger Graph K_n, wenn er regulär vom Grad $n - 1$ ist.

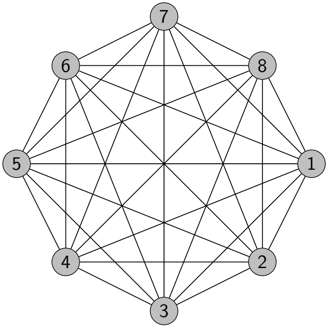
Bemerkung: In einem vollständigen Graphen sind somit alle Knoten direkt miteinander verbunden.

Eigenschaft: Jeder K_n hat genau $\frac{n \cdot (n-1)}{2}$ Kanten:

$$|E| = \frac{n \cdot (n-1)}{2}$$

Vollständiger Graph (2)

Beispiel: Vollständiger Graph $G = K_8$



Komplementärgraph (1)

Sei $G = (V, E)$ ein schlichter Graph, und K die Menge aller zwei-elementiger Teilmengen von V

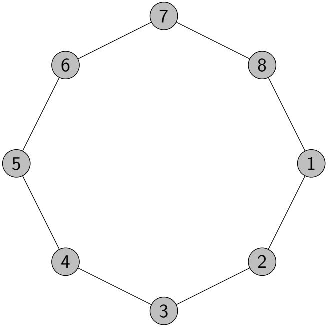
Definition (Komplementärer Graph)

$G' = (V, K \backslash E)$ ist der komplementäre Graph zu $G = (V, E)$.

Bemerkung: Der komplementäre Graph G' besitzt also eine Kante zwischen den Knoten i und j *genau dann*, wenn sie *nicht* im ursprünglichen Graphen G enthalten ist.

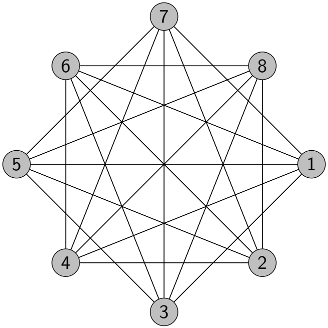
Komplementärgraph (2)

Beispiel: Graph G



Komplementärgraph (3)

Beispiel: der komplementäre Graph G' zu G sieht folgendermaßen aus:



Bipartite Graphen (1)

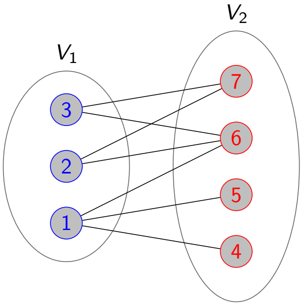
Definition (Bipartiter Graph)

Ein Graph $G = (V, E)$ heißt *bipartit* wenn eine Partitionierung der Knoten V in zwei disjunkte Teilmengen V_1 und V_2 existiert, sodaß für jede Kante $[i, j] \in E$ gilt, dass entweder $i \in V_1$ und $j \in V_2$, oder andererseits $i \in V_2$ und $j \in V_1$.

Bemerkung: Man stellt sich am besten eine Knotenmenge auf der linken Seite, und eine auf der rechten Seite vor. Kanten verlaufen nur von links nach rechts (und umgekehrt), aber niemals zwischen Knoten einer der beiden Mengen.

Bipartite Graphen (2)

Beispiel:



Teilgraphen (1)

Definition (Teilgraph)

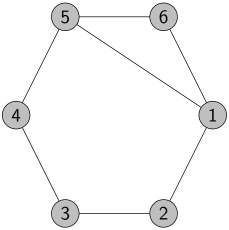
Ein Graph $G' = (V', E')$ heißt *Teilgraph*, *Subgraph* oder *Untergraph* von $G = (V, E)$, wenn $V' \subseteq V$ und $E' \subseteq E$ (und für alle $[i, j] \in E'$ gilt $i \in V'$ und $j \in V'$)

Definition (Obergraph)

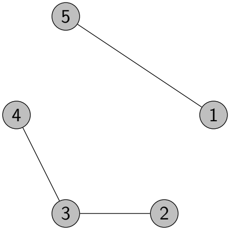
Ein Graph G ist *Obergraph* eines Graphen G' wenn G' Teilgraph von G ist.

Teilgraphen (2)

Beispiel: Graph G



Beispiel: Teilgraph G' von G



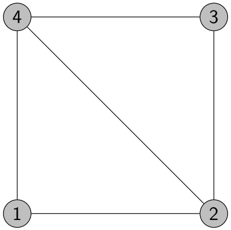
Unterteilungsgraph

Definition (Unterteilungsgraph)

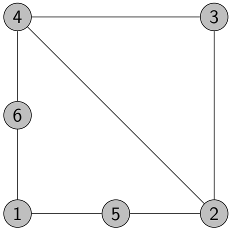
Ein Unterteilungsgraph ist ein Graph, der durch Kantenunterteilung aus einem anderen Graph entstanden ist.

Anmerkung: In einem Unterteilungsgraphen wird "auf einer Kante" ein neuer Knoten eingefügt. Formal wird eine Kante $[u, v]$ entfernt, ein neuer Knoten w dem Graphen hinzugefügt, und dann Kanten $[u, w]$ und $[w, v]$ hinzugefügt.

Beispiel: Graph G



Beispiel: Unterteilungsgraph von G



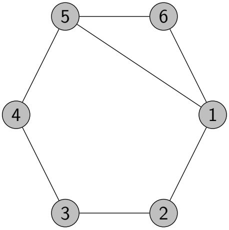
Spannender Teilgraph

Definition (Spannender Teilgraph)

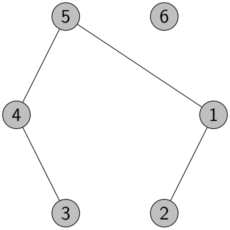
Ein Teilgraph $G' = (V', E')$ heißt *spannender Teilgraph* von $G = (V, E)$ wenn $V' = V$ und $E' \subseteq E$.

Bemerkung: Ein spannender Teilgraph G' besitzt also alle Knoten von G, jedoch nicht unbedingt alle Kanten von G (eventuell auch gar keine).

Beispiel: Graph G



Beispiel: Teilgraph G' von G



Gesättigter Teilgraph (1)

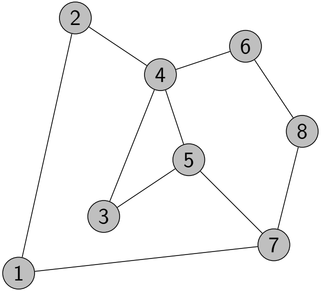
Definition (Gesättigter Teilgraph)

Sei $G' = (V', E')$ ein Teilgraph von $G = (V, E)$. Enthält E' alle Kanten aus E deren Endpunkte in V' liegen, so sagt man, dass G' von $V' \subseteq V$ in G aufgespannt wird, oder dass G' ein *gesättigter Teilgraph* von G ist.

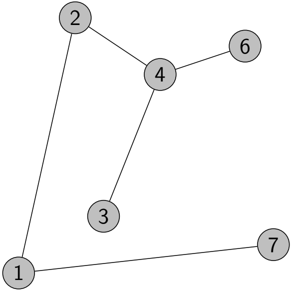
Bemerkung: Ein gesättigter Teilgraph enthält also bezüglich einer Teilmenge von Knoten alle Kanten des ursprünglichen Graphen. Es fehlen also alle Kanten, die zu den fehlenden Knoten inzident sind.

Gesättigter Teilgraph (2)

Graph G mit 8 Knoten



Gesättigter Teilgraph G' mit 6 Knoten von G



Kantenfolgen

Definition (Kantenfolge)

Eine Kantenfolge von v_1 nach v_n (aus V) ist eine endliche Folge von Kanten $[v_1, v_2], [v_2, v_3], \ldots, [v_{n-1}, v_n]$, sodass jeweils zwei aufeinanderfolgende Kanten adjazent sind.

Definition (Offene Kantenfolge)

Eine Kantenfolge $[v_1, v_2], \ldots, [v_{n-1}, v_n]$ heißt *offen*, falls $v_1 \neq v_n$.

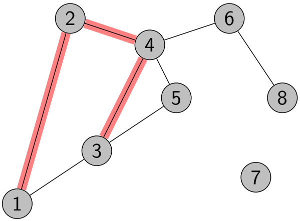
Definition (Geschlossene Kantenfolge)

Eine Kantenfolge $[v_1, v_2], \ldots, [v_{n-1}, v_n]$ heißt *geschlossen*, falls $v_1 = v_n$.

Kantenfolgen

Bemerkung: In einer Kantenfolge können bestimmte Kanten auch mehrfach vorkommen!

Beispiel: Kantenfolge $K = [1,2],[2,4],[4,3],[3,4],[4,2]$



Beispiel: *Keine* Kantenfolgen wären $[1,2],[2,3]$ oder $[1,2],[4,3]$.

Kantenzug

Definition (Kantenzug)

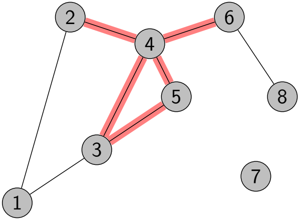
Ein *Kantenzug* ist eine Kantenfolge bei der alle Kanten paarweise verschieden sind.

Anmerkung 1: "paarweise verschieden" ... typische (mathematische) Formulierung, die ausdrückt, dass alle Elemente unterschiedlich sind, d.h. es keine zwei gleichen Elemente gibt!

Anmerkung 2: Analog zur Kantenfolge existieren *offene* und *geschlossene* Kantenzüge

Kantenzug

Beispiel: Kantenzug $K = [2,4],[4,5],[5,3],[3,4],[4,6]$



Anmerkung: der Knoten 4 wird in diesem Kantenzug mehrfach besucht!

Weg, bzw. Pfad (1)

Definition (Weg, Pfad)

Ein Weg oder Pfad (engl. path) $P(s, t)$ ist ein offener Kantenzug von $s \in V$ nach $t \in V$ bei dem alle Knoten paarweise verschieden sind.

Achtung

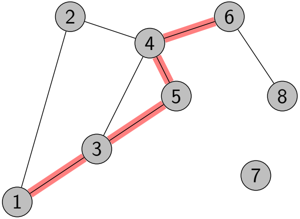
In der Literatur wird oft *Weg* so definiert, daß gleiche Knoten im Kantenzug vorkommen dürfen. bei einem *Pfad* müssen dann die Knoten voneinander verschieden sein. Allerdings existieren auch andere Definitionen in welchen dieser "Weg" als *Pfad* bezeichnet wird, und der zuvor genannte "Pfad" als *elementarer Pfad*.

Definition (Weglänge)

Die Länge $|P(s, t)|$ ist die Anzahl der Kanten in $P(s, t)$

Weg, bzw. Pfad (2)

Beispiel: der Pfad $P(1,6) = [1,3],[3,5],[5,4],[4,6]$ mit $|P(1,6)| = 4$ ist rot markiert



Allgemein können mehrere (verschiedene) Wege von s nach t existieren.

Definition (Kürzester Weg/Pfad)

Jene Wege von s nach t mit minimaler Länge $|P(s,t)|$
werden **kürzeste Wege** (oder kürzeste Pfade) genannt.

Kreis, Zyklus

Definition (Zyklus)

Ein *Zyklus* $(v_1, v_2, \ldots, v_{n-1}, v_n)$ ist ein geschlossener Kantenzug, d.h. $v_1 = v_n$.

Definition (Kreis)

Ein *Kreis* ist ein Zyklus bei dem alle Knoten, bis auf den Start- und Endknoten verschieden sind.

Achtung

Auch die Begriffe Kreis und Zyklus sind leider in der Literatur nicht immer einheitlich definiert.

Eulersche Linie, Euler-Zyklus

Definition (Eulersche Linie)

Eine *Eulersche Linie* ist ein Kantenzug, der alle Kanten des Graphen genau einmal enthält.

Bemerkung: Man beachte, daß hier Knoten mehrfach durchlaufen werden können!

Definition (Euler-Zyklus)

Bei einer geschlossenen *Eulerschen Linie*, bzw. ein **Euler-Zyklus** sind Start- und Endknoten gleich!

Achtung

Oft wird von Eulerschen Kreisen oder Eulerschen Zyklen gesprochen. Ob diese Bezeichnungen tatsächlich zutreffend sind hängt wiederum von den konkreten Definitionen von diesen Begriffen ab, also insbesondere ob sie das mehrfache Durchlaufen von Knoten erlauben.

Eulersche Graphen

Definition (Eulerscher Graph)

Besitzt ein Graph einen *Euler-Zyklus*, so bezeichnet man ihn als *Eulersch*.

Satz von Euler-Hierholzer

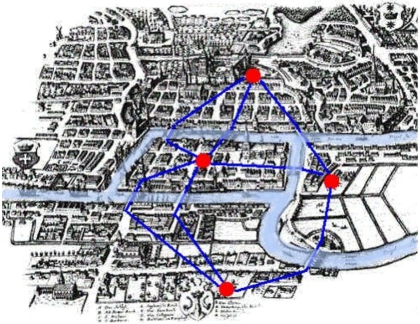
Sei G ein zusammenhängender Graph. Dann sind folgende Aussagen äquivalent:

- G ist *Eulersch*
- Jeder Knoten in G hat geraden Grad
- Die Kantenmenge von G ist die Vereinigung aller Kanten von paarweise disjunkten Zyklen [a]

[a] disjunkt bezüglich ihrer Kanten!

Königsberger Brückenproblem

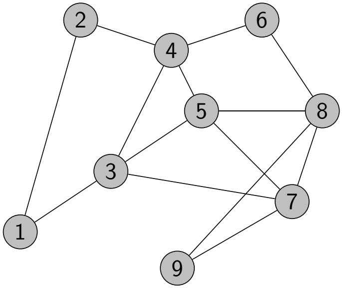
Wir haben nun die allgemeine Lösung des eingangs erwähnten Königsberger Brückenproblems



Anmerkung: Ein Graph besitzt genau dann eine offene Eulersche Linie, wenn alle Knoten bis auf zwei einen geraden Grad aufweisen.

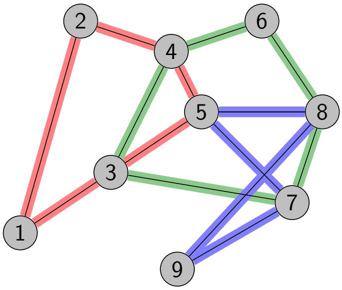
Euler-Zyklus

Beispiel: Beispiel, indem die Kantenmenge von G die Vereinigung von paarweise disjunkten Zyklen ist (und damit G eulersch)



Euler-Zyklus

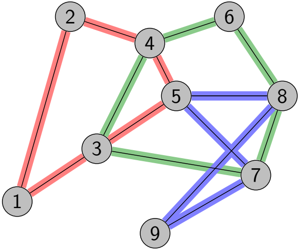
Beispiel: Beispiel, indem die Kantenmenge von G die Vereinigung von paarweise disjunkten Zyklen ist (und damit G eulersch)



Algorithmus von Hierholzer

1. Suche Zyklus Z in Graphen G
2. Ist Z ein Euler-Zyklus $\Rightarrow$ fertig
3. Wähle einen Knoten $v \in Z$ mit $d(v) > 0$ wobei die Kanten von Z ignoriert werden und suche neuen Zyklus Z'
4. Füge Z' mit Z zusammen indem Z' am ersten Schnittpunkt mit Z zur Gänze eingefügt wird, danach der Rest von Z
5. Dieser neue Zyklus wird nun Z genannt. Weiter mit Schritt (2)

Euler-Zyklus



1. $Z = (1, 2, 4, 5, 3, 1)$
2. $Z' = (4, 6, 8, 7, 3, 4)$
3. Kombiniere Z mit Z' zu $Z = (1, 2, 4, 6, 8, 7, 3, 4, 5, 3, 1)$
4. $Z' = (5, 8, 9, 7, 5)$
5. Kombiniere Z mit Z' zu $Z = (1, 2, 4, 6, 8, 7, 3, 4, 5, 8, 9, 7, 5, 3, 1)$

Hamiltonsche Linie/Kreis

Definition (Hamiltonsche Linie)

Eine *Hamiltonsche Linie* ist ein Weg, der alle Knoten des Graphen *genau* einmal enthält.

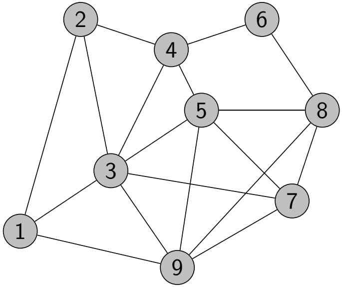
Bemerkung: In einer Hamiltonsche Linie dürfen die Knoten nicht mehrfach (wiederholt) besucht werden!

Definition (Hamilton-Kreis)

Ein *Hamilton-Kreis* ist ein Kreis der alle Knoten des Graphen genau einmal enthält.

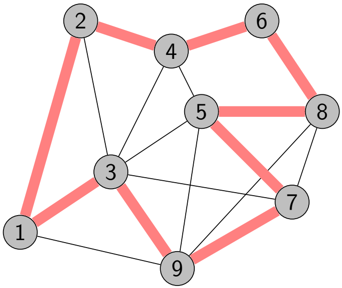
Hamiltonkreis

Beispiel:



Hamiltonkreis

Beispiel:



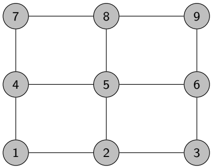
Hamiltonscher Graph

Definition (Hamiltonscher Graph)

Einen Graphen der einen Hamiltonkreis enthält nennt man *Hamiltonschen Graph*.

- Nicht alle Graph sind Hamiltonsch!
- **Es existiert keine einfache Charakterisierung ob ein Graph Hamiltonsch ist!**
- Im Gegensatz zum leicht lösbaren Problem des Euler-Zyklus, ist das Hamiltonkreisproblem im Allgemeinen sehr schwierig zu lösen.

Beispiel:



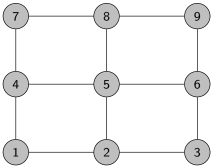
Hamiltonscher Graph

Definition (Hamiltonscher Graph)

Einen Graphen der einen Hamiltonkreis enthält nennt man *Hamiltonschen Graph*.

- Nicht alle Graph sind Hamiltonsch!
- **Es existiert keine einfache Charakterisierung ob ein Graph Hamiltonsch ist!**
- Im Gegensatz zum leicht lösbaren Problem des Euler-Zyklus, ist das Hamiltonkreisproblem im Allgemeinen sehr schwierig zu lösen.

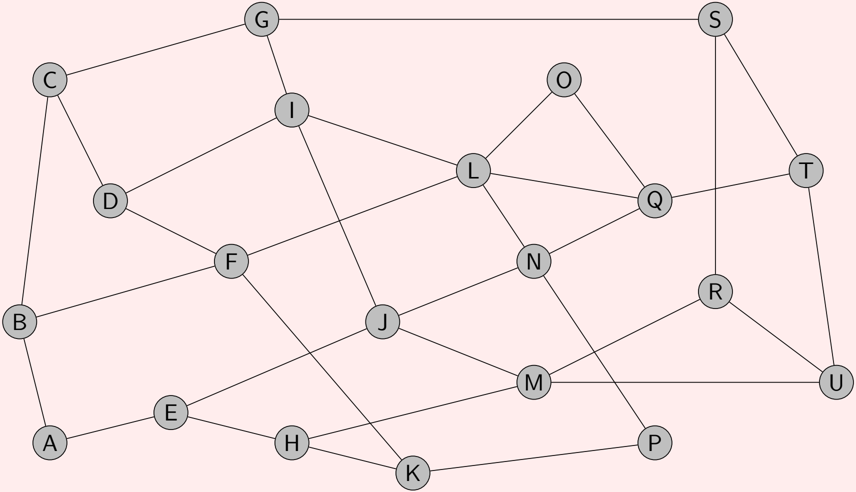
Beispiel:



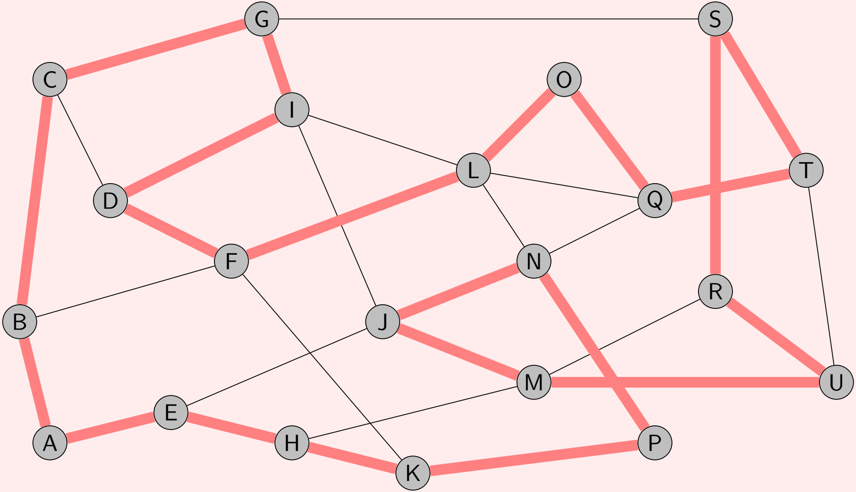
Dieser Graph ist *nicht* Hamiltonsch!

Übungsaufgabe zu Hamilton-Kreis

Geben Sie einen Hamilton-Kreis für den folgenden Graphen an.



Lösung



Distanz (ungewichteter Graph)

Definition (Distanz)

Die Distanz (Abstand) zweier Knoten u und v in einem Graphen ist die Länge des kürzesten Weges $P(u, v)$, also

$$d(u, v) = \min_{P(u,v)} |P(u, v)|.$$

- Falls kein Weg zwischen u und v existiert, so gilt

$$d(u, v) = \infty$$

- Der Abstand eines Knoten zu sich selbst ist 0, also

$$d(u, v) = 0 \leftrightarrow u = v.$$

- In ungerichteten Graphen gilt $d(u, v) = d(v, u)$ für alle $u, v \in V$.

Distanz (gewichteten Graphen)

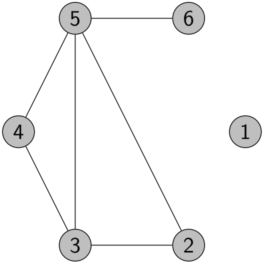
In einem gewichteten Graphen gilt für die Distanz

$$d(u, v) = \min_{P(u,v)} \sum_{e \in P(u,v)} w(e),$$

also die Summe der Kantengewichte über den kürzesten Pfad von u nach v. Die folgenden Definitionen können für beide Varianten (ungewichtete und gewichtete Graphen) verwendet werden.

Distanz (Beispiel)

Graph G mit 6 Knoten



$d(1,1) = 0$

$d(1,2) = \infty$

$d(2,6) = 2$

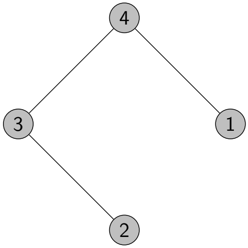
$d(3,4) = 1$

Exzentrizität

Definition (Exzentrizität)

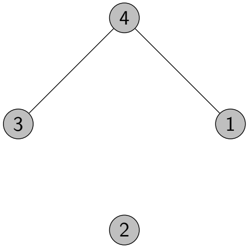
Die Exzentrizität $ex(v)$ eines Knoten $v \in V$ eines ungerichteten, *zusammenhängenden* Graphen $G = (V, E)$ ist die Distanz zum entferntesten Knoten von v in G

Zur Ermittlung der Exzentrizitäten muss der maximale Abstand jedes Knoten im Graph G bestimmt werden.



Knoten	Exzentrizität	Entferntester Knoten
1	3	2
2	3	1
3	2	1
4	2	2

Exzentrizität (2)



In einem nicht zusammenhängenden Graphen gibt es zu jedem Knoten mindestens einen Knoten der nicht erreichbar ist.
Der Wert für die Exzentrizität ist in diesem Fall für alle Knoten mit ∞ definiert.

$$\forall v \in V : ex(v) = \infty$$

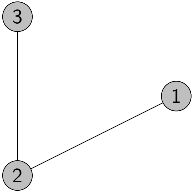
Durchmesser

Definition (Durchmesser)

Der Durchmesser $dm(G)$ eines Graphen G ist das Maximum der Exzentrizitäten aller Knoten von G.

Wenn G nicht zusammenhängend ist, so gilt $dm(G) = \infty$.

Beispiel: Graph mit $dm(g) = 2$



Knoten	Exzentrizität
1	2
2	1
3	2

Anmerkung: Das Maximum aller Exzentrizitäten im Beispielgraphen beträgt 2. Somit beträgt der Durchmesser $dm(G) = 2$

Radius (1)

Definition (Radius)

Der Radius $rad(G)$ eines Graphen G ist das Minimum aller Exzentrizitäten aller Knoten von G.

Für alle ungerichteten zusammenhängenden Graphen gilt:

$$rad(G) \leq dm(G) \leq 2 \cdot rad(G)$$

sowie

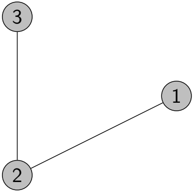
$$dm(G) \geq rad(G) \geq {}^{dm(G)}/_{2}$$

Anmerkung: Das heißt, der Radius kann maximal gleich groß sein wie der Durchmesser und muss mindestens halb so groß wie der Durchmesser sein.

Anmerkung: Wenn der Graph nicht zusammenhängend ist, ist der Radius unendlich ($rad(G) = \infty$).

Radius (2)

Beispiel: Graph mit $rad(G) = 1$



Knoten	Exzentrizität
1	2
2	1
3	2

Anmerkung: Das Minimum aller Exzentrizitäten im Beispielgraphen beträgt 1. Somit beträgt der Radius
$rad(G) = 1$

Zentrum (1)

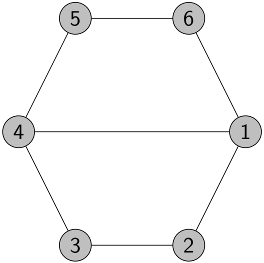
Definition (Zentrum)

In einem ungerichteten, zusammenhängenden Graphen G ist das Zentrum $Z(G)$ die Menge der Knoten $v \in V$ deren Exzentrizität dem Radius entspricht:

$$Z(G) = \{v \in V(G) : ex(v) = rad(G)\}$$

Zentrum (2)

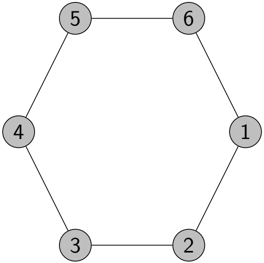
Beispiel: Graph G_1 mit 6 Knoten



Somit gilt: $dm(G_1) = 3, rad(G_1) = 2, Z(G_1) = \{1, 4\}$

Zentrum (3)

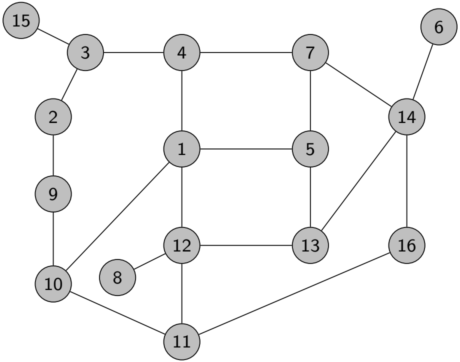
Beispiel: Graph G_2 mit 6 Knoten



Somit gilt: $dm(G_2) = 3, rad(G_2) = 3, Z(G_2) = \{1, 2, 3, 4, 5, 6\}$

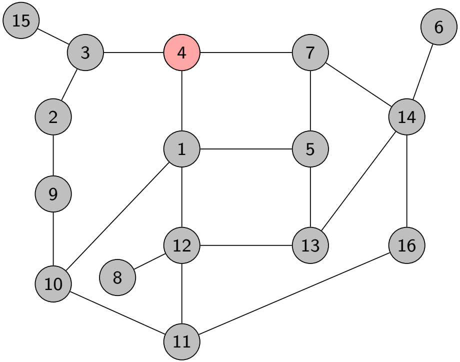
Zentrum (4)

Beispiel: In einer Stadt soll eine neue Feuerwache gebaut werden. Um im Brandfall eine optimale Anfahrtszeit zu gewährleisten, soll der Standort der Feuerwache so gewählt werden, dass auch das am weitesten entfernte Haus schnell zu erreichen ist.
Der Graph stellt die Stadt dar. Ein Haus wird durch einen Knoten und eine Straßenverbindung durch eine Kante symbolisiert. Aus diesem Graph lässt sich sehr rasch das Zentrum der Stadt berechnen.



Zentrum (4)

Beispiel: In einer Stadt soll eine neue Feuerwache gebaut werden. Um im Brandfall eine optimale Anfahrtszeit zu gewährleisten, soll der Standort der Feuerwache so gewählt werden, dass auch das am weitesten entfernte Haus schnell zu erreichen ist.
Der Graph stellt die Stadt dar. Ein Haus wird durch einen Knoten und eine Straßenverbindung durch eine Kante symbolisiert. Aus diesem Graph lässt sich sehr rasch das Zentrum der Stadt berechnen.



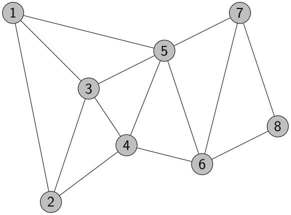
v	$ex(v)$	v	$ex(v)$
1	4	9	5
2	5	10	4
3	4	11	5
4	3	12	4
5	4	13	5
6	5	14	4
7	4	15	5
8	5	16	5

Zusammenhang

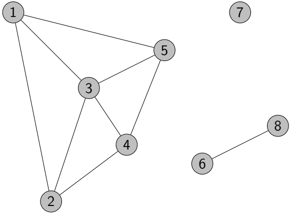
Definition (Zusammenhang)

Ein ungerichteter Graph $G = (V, E)$ heißt genau dann *zusammenhängend*, wenn für alle Paare $u, v \in V$ gilt, dass sie durch einen Weg verbunden sind (kurz: $u \rightsquigarrow v, \forall u, v \in V(G)$).

Beispiel: Zusammenhängender Graph G_1:



Beispiel: *Nicht* zusammenhängender Graph G_2:



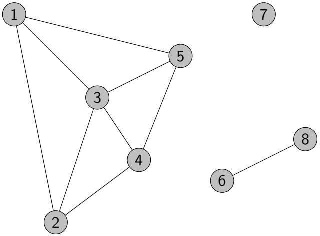
Zusammenhangskomponenten

Definition (Zusammenhangskomponenten)

Unter einer *Zusammenhangskomponenten* $K(v)$ versteht man eine *Teilmenge* von Knoten *maximaler Größe* die mit v durch einen Weg verbunden sind.

$$K(v) = \{u \in V(G) \mid v \rightsquigarrow u\}$$

Beispiel: Graph G_3 mit 3 Zusammenhangskomponenten:



$$Z_1 = \{1, 2, 3, 4, 5\}$$

$$Z_2 = \{6, 8\}$$

$$Z_3 = \{7\}$$

Komponenten

Definition (Komponenten)

Die von den Zusammenhangskomponenten aufgespannten, gesättigten Teilgraphen von G heißen die *Komponenten* des Graphen G.

Definition (Anzahl der Komponenten)

Die *Anzahl* der Komponenten eines Graphen G wird mit $c(G)$ bezeichnet.

Beispiel: $c(G_3) = 3$

Beispiel: Die Komponenten von G_3 lauten:

$K_1 = (\{1, 2, 3, 4, 5\}, \{[1, 2], [1, 3], [1, 5], [2, 3], [2, 4], [3, 4], [3, 5], [4, 5]\})$

$K_2 = (\{6, 8\}, \{[6, 8]\})$

$K_3 = (\{7\}, \emptyset)$

Artikulation

Definition (Artikulation)

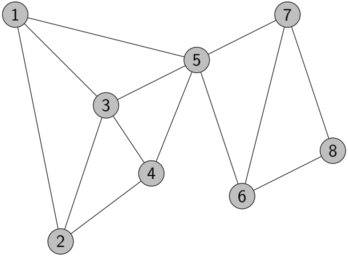
Ein Knoten v heißt *Artikulation* wenn die Anzahl der Komponenten von $G - \{v\}$ größer ist als jene von G, also

$$c(G - \{v\}) > c(G).$$

- Artikulationen haben niemals Grad 0 oder 1.
- Artikulationen sind Schnittstellen zwischen größer gleich zwei Blöcken.
- Nach Entfernung einer Artikulation ist der Graph nicht (mehr) zusammenhängend.

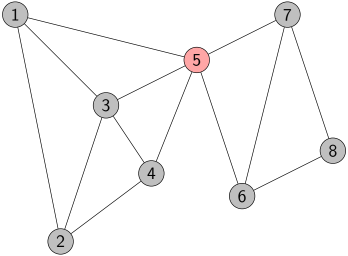
Artikulation

Beispiel: Graph G_4:



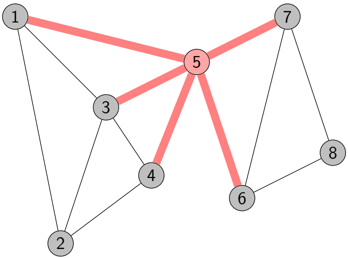
Artikulation

Beispiel: Graph G_4:



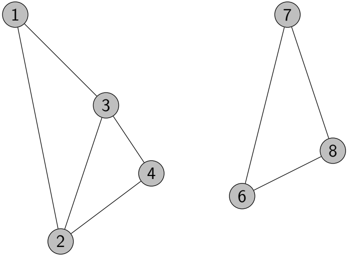
Artikulation

Beispiel: Graph G_4:



Artikulation

Beispiel: Graph G_4:



$G - \{5\}$ besteht nun aus den Komponenten
$K_1 = \big(K(1), \{[1,2], [1,3], [2,3], [2,4], [3,4]\}\big)$ und
$K_2 = \big(K(6), \{[6,7], [6,8], [7,8]\}\big)$.

Blöcke

Definition (Block)

Ein *Block* ist ein zusammenhängender Teilgraph, der keine Artikulationen hat und es keinen Obergraph zu diesem Teilgraph gibt, der ebenfalls keine Artikulationen hat.

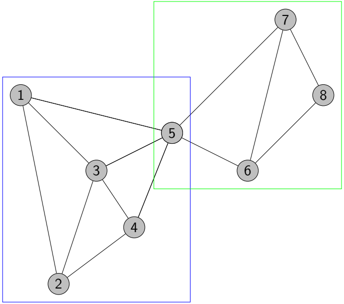
Bemerkung: ein Block ist also ein *maximaler* Teilgraph ohne Artikulationen (maximal bezüglich der Anzahl seiner Knoten).

Folgerungen:

- Jede Kante und jeder Kreis liegen in genau einem Block von G.
- Es gibt keine Kante die in zwei Blöcken liegen kann.
- Zwei Blöcke eines Graphen haben höchstens einen gemeinsamen Knoten und dieser ist eine Artikulation.
- Der kleinste Block eines Graphen ist ein isolierter Knoten.

Blöcke

Beispiel: Graph G_5 mit zwei Blöcken:



Die Knotenmengen der Blöcke lauten:

$B_1 = \{1, 2, 3, 4, 5\}$
$B_2 = \{5, 6, 7, 8\}$

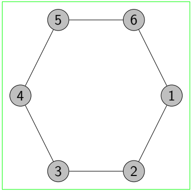
Der Block selbst ist der durch diese Knotenmengen definierte spannende, gesättigte Teilgraph.

Blöcke

Beispiel: Der Kleinstmögliche Block besteht nur aus einem Knoten. Graph G_6:

Knotenmenge zum Block: $B_1 = \{1\}$

Beispiel: Ein zusammenhängender Graph ohne Artikulation ist selbst ein Block. Graph G_7:



Knotenmenge zum Block: $B_1 = \{1, 2, 3, 4, 5, 6\}$

Brücke

Definition (Brücke)

Eine Kante e eines Graphen G heißt *Brücke*, wenn sich nach Entfernung dieser Kante die Anzahl der Komponenten des Graphen erhöht, also

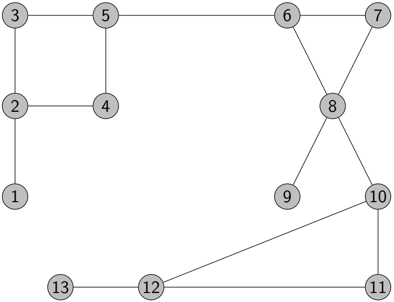
$$c(G - \{e\}) > c(G).$$

Anmerkung:

- Eine Brücke ist selbst ein Block.
- Eine Brücke ist damit, wie auch eine Artikulation, eine Schwachstelle im Graphen.

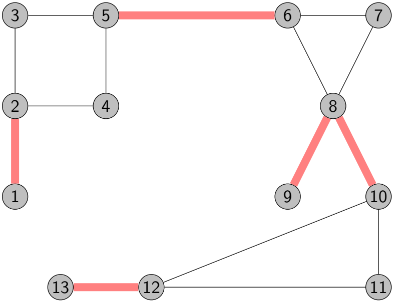
Brücken

Beispiel: Graph G_8 mit 5 Brücken



Brücken

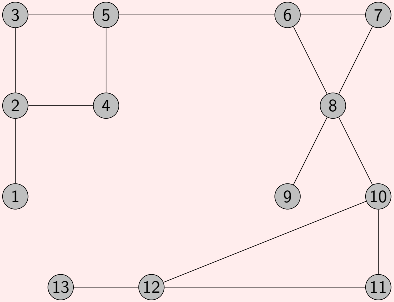
Beispiel: Graph G_8 mit 5 Brücken



Brücken

Beispiel 6.01

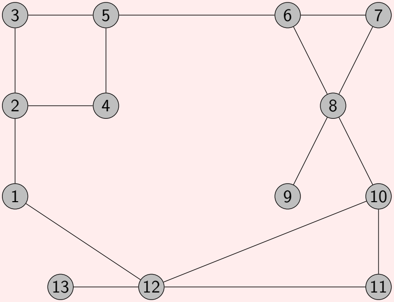
Gegeben sei der folgende Graph G_8. Bestimmen Sie alle Artikulationen, Brücken und Blöcke.



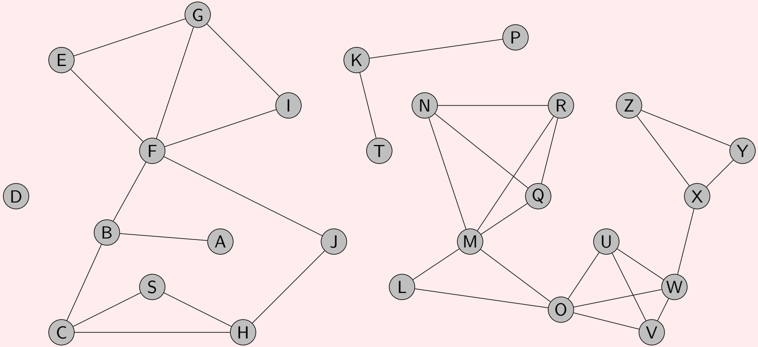
Brücken

Beispiel 6.02

Gegeben sei der folgende Graph G_9. Bestimmen Sie alle Artikulationen, Brücken und Blöcke.



Beispiel 6.11: Gegeben sei der folgende Graph G_{10}:

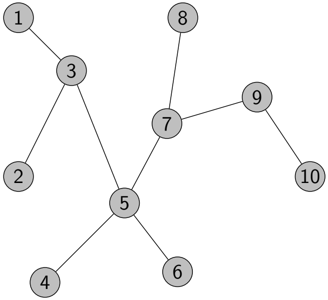


1. Geben Sie alle Zusammenhangskomponenten an.
2. Bestimmen Sie alle Artikulationen.
3. Bestimmen Sie alle Brücken.
4. Bestimmen Sie alle Blöcke, und schreiben Sie diese jeweils als Knotenmenge auf.

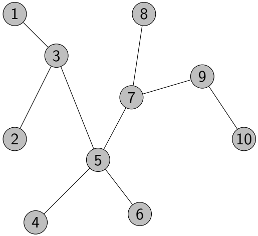
Bäume

Definition (Baum)

Ein zusammenhängender Graph der keine Kreise enthält wird *Baum* T genannt.



Bäume



Ein Baum T kann durch folgende äquivalente Aussagen charakterisiert werden:

- T ist ein Baum
- T ist zusammenhängend und kreisfrei
- Zwei beliebige Knoten von T sind durch genau einen Weg verbunden
- T hat $n - 1$ Kanten und ist zusammenhängend
- T hat $n - 1$ Kanten und ist kreisfrei
- T ist zusammenhängend und jede Kante ist eine Brücke

Begriffe

- **Grad eines Baumes:** maximaler Grad eines Knoten des Baumes
- **Blatt:** Knoten mit Grad 1 heißen *Blätter* des Baumes. Blätter haben nur einen Nachbarn. Jeder Baum hat mindestens ein Blatt.
- **Ast:** Die Kanten eines Baumes werden auch als *Äste* bezeichnet.
- **Innerer Knoten:** Ein Knoten heißt *innerer Knoten* wenn er kein Blatt ist.

Anmerkung: Bäume haben zahlreiche Anwendungen in der Informatik als Datenstrukturen, z.B. können Objekte in Bäumen so abgespeichert werden, dass sie schnell gesucht (gefunden) werden können!

Wurzelbäume, Arboreszenzen

Definition (Wurzelbaum, Arboreszenz)

Ein gerichteter Graph G heißt *Wurzelbaum* oder *Arboreszenz*, wenn er

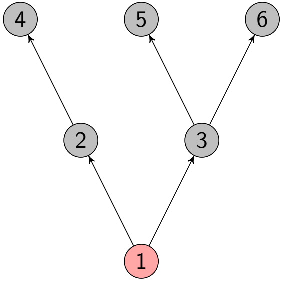
- zusammenhängend ist, und
- es genau einen Knoten $w \in V(G)$ gibt mit $d^-(w) = 0$, und
- für alle anderen Knoten $v \in V(G)$ gilt $d^-(v) = 1$.

Hierbei bezeichnet $d^-(v)$ den *Eingangsgrad* (engl. in-degree), also die Anzahl der zum Knoten führenden Kanten. Der *Ausgangsgrad* $d^+(v)$ (engl. out-degree) bezeichnet hingegen die vom Knoten wegführenden Kanten.

Anmerkung: Ein Wurzelbaum ist zusammenhängend und hat genau einen Knoten mit Eingangsgrad 0. Jeder andere Knoten hat den Eingangsgrad 1. Eine Arboreszenz ist ein gerichteter Graph dessen Schatten ein Baum ist, sodass ein Knoten w ausgezeichnet ist (Wurzel), und jede Kante der Arboreszenz von w weggerichtet ist.

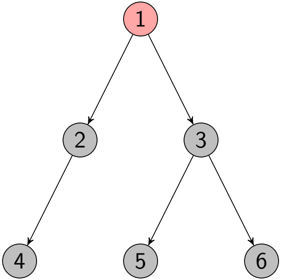
Wurzelbäume, Arboreszenzen

Beispiel: Ursprünglich wurden Wurzelbäume so gezeichnet:



Wurzelbäume, Arboreszenzen

Beispiel: Heute wird die Wurzel immer oben gezeichnet.



Binärbaum

Ein *Binärbaum* ist ein (gerichteter) Wurzelbaum, in dem jeder Knoten höchstens zwei **Kinder** besitzt. Manchmal wird eine explizite Unterscheidung zwischen linkem und rechtem Kind vorgenommen. Wir unterscheiden:

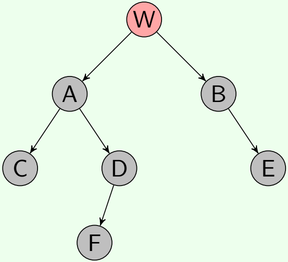
- **Voller Binärbaum** (auch: *strikt* oder *saturiert* gennant): Jeder innere Knoten besitzt *genau* zwei Kinder. (Ein Blatt hat keine Kinder.)
- **Vollständiger Binärbaum**: Alle Ebenen sind gefüllt, außer evtl. die letzte, die von links nach rechts gefüllt wird.
- **Perfekter Binärbaum**: Ist sowohl voll als auch vollständig; dann besitzt die letzte Ebene genau 2^h Blätter.

Remark: Literaturhinweis: Manche Quellen verwenden die Begriffe *voll* und *vollständig* synonym oder vertauscht. Hier gilt: "voll" $\Rightarrow$ jeder innere Knoten hat zwei Kinder; "vollständig" $\Rightarrow$ linkslückenlose Füllung der Ebenen.

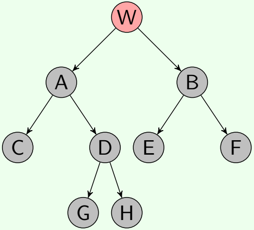
Arten von Binärbäumen

Beispiele

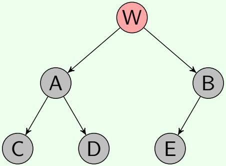
Allgemeiner Binärbaum



Voller, aber nicht vollständig



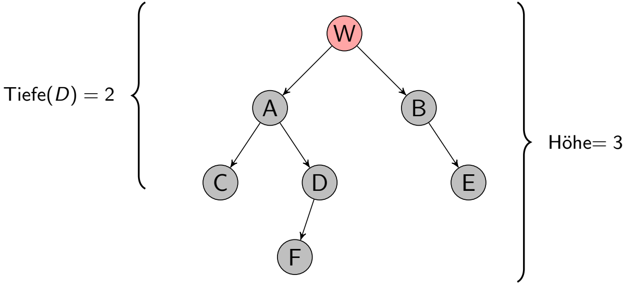
Vollständig, aber nicht voll



Links: jeder innere Knoten hat zwei Kinder (voll); mittig: volle Struktur, aber letzte Ebene nicht lückenlos (nicht vollständig); rechts: Ebenen füllen sich lückenlos von links, aber Knoten B hat nur ein Kind (nicht voll).

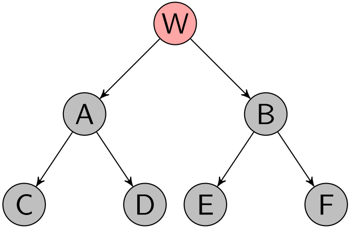
Ein vollständiger und voller Binärbaum wird *perfekter* Binärbaum genannt.

Höhe und Tiefe eines Baumes



- Tiefe(v) = Anzahl Kanten von der Wurzel
- Höhe(T) = maximale Tiefe über alle Knoten

Zusammenhang Höhe und Anzahl an Blättern



Ebene 0: 1 Knoten $= 2^0$ Knoten

Ebene 1: 2 Knoten $= 2^1$ Knoten

Ebene 2: 4 Knoten $= 2^2$ Knoten

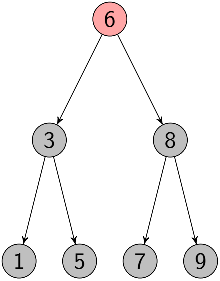
- Ein perfekter Binärbaum mit Höhe h hat 2^h Blätter und $2^{(h+1)} - 1$ Knoten.
- Ein perfekter Binärbaum mit n Blättern hat $h = \mathrm{ld}\ n$.
- Dabei steht ld für den *Logarithmus Dualis*, also den Logarithmus zur Basis 2.

Suchbäume

- Bäume eignen sich zum effizienten Suchen, Einfügen und Löschen von Elementen.
- Beim Einfügen von Elementen in den Suchbaum wird der Baum so erweitert, dass für jeden Knoten gilt: Alle Werte im linken Teilbaum sind kleiner, alle Werte im rechten Teilbaum sind größer oder gleich dem Wert des Knotens.
- Die Struktur des Baumes wird beim Einfügen (zunächst) nicht verändert.
- Später betrachten wir balancierte Suchbäume, wo dies sehr wohl geschieht.

(Binärer) Suchbaum

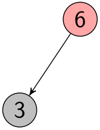
- Achtung: in der Darstellung ist in den Knoten der *Inhalt* (also ein `int`-Wert) dargestellt (und nicht der Knotenname, bzw. Index)



- Binäre Suchbäume:
  - Ein vollständiger voller Binärbaum mit Höhe d hat $2^d - 1$ Knoten
  - Beispiel: Höhe 3 $\Rightarrow 2^3 - 1 = 7$ Knoten
  - Umgekehrt hat ein vollständiger und voller Binärbaum mit n Knoten eine Höhe von $O(\mathrm{ld}\ n)$

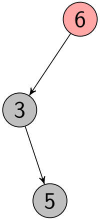
- Schrittweiser Aufbau eines Suchbaumes
- Einfügen von Elementen 6, 3, 5, 8, 1, 7, 9, 4:
  - Das erste Element “6” wird als Wurzelknoten in den Baum aufgenommen.

- Schrittweiser Aufbau eines Suchbaumes
- Einfügen von Elementen 6, 3, 5, 8, 1, 7, 9, 4:



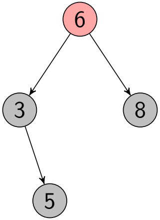
- Das erste Element “6” wird als Wurzelknoten in den Baum aufgenommen.
- Da $3 < 6$ kommt 3 als linker Nachfolger (“Kind”) der Wurzel.

- Schrittweiser Aufbau eines Suchbaumes
- Einfügen von Elementen 6, 3, 5, 8, 1, 7, 9, 4:



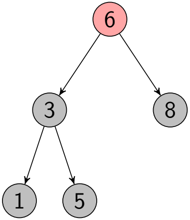
- Das erste Element "6" wird als Wurzelknoten in den Baum aufgenommen.
- Da $3 < 6$ kommt 3 als linker Nachfolger ("Kind") der Wurzel.
- $5 < 6$ also kommt 5 in den linken Teilbaum. Dort befindet sich ein Knoten mit Inhalt 3. Da $5 > 3$ wird 5 als rechtes Kind von 3 in den Baum eingefügt.

- Schrittweiser Aufbau eines Suchbaumes
- Einfügen von Elementen 6, 3, 5, 8, 1, 7, 9, 4:



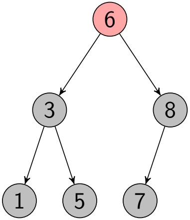
- Das erste Element "6" wird als Wurzelknoten in den Baum aufgenommen.
- Da $3 < 6$ kommt 3 als linker Nachfolger ("Kind") der Wurzel.
- $5 < 6$ also kommt 5 in den linken Teilbaum. Dort befindet sich ein Knoten mit Inhalt 3. Da $5 > 3$ wird 5 als rechtes Kind von 3 in den Baum eingefügt.
- $8 > 6$, das rechte Kind von der Wurzel existiert noch nicht, also kommt 8 dort hin!

- Schrittweiser Aufbau eines Suchbaumes
- Einfügen von Elementen 6, 3, 5, 8, 1, 7, 9, 4:



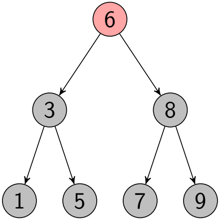
- Das erste Element "6" wird als Wurzelknoten in den Baum aufgenommen.
- Da $3 < 6$ kommt 3 als linker Nachfolger ("Kind") der Wurzel.
- $5 < 6$ also kommt 5 in den linken Teilbaum. Dort befindet sich ein Knoten mit Inhalt 3. Da $5 > 3$ wird 5 als rechtes Kind von 3 in den Baum eingefügt.
- $8 > 6$, das rechte Kind von der Wurzel existiert noch nicht, also kommt 8 dort hin!
- 1 wird im linken Teilbaum am ersten freien Platz eingefügt, also als linkes Kind von 3.

- Schrittweiser Aufbau eines Suchbaumes
- Einfügen von Elementen 6, 3, 5, 8, 1, 7, 9, 4:



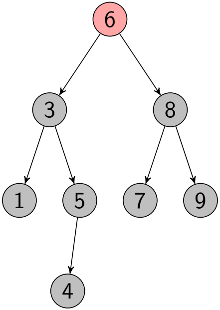
- Das erste Element "6" wird als Wurzelknoten in den Baum aufgenommen.
- Da $3 < 6$ kommt 3 als linker Nachfolger ("Kind") der Wurzel.
- $5 < 6$ also kommt 5 in den linken Teilbaum. Dort befindet sich ein Knoten mit Inhalt 3. Da $5 > 3$ wird 5 als rechtes Kind von 3 in den Baum eingefügt.
- $8 > 6$, das rechte Kind von der Wurzel existiert noch nicht, also kommt 8 dort hin!
- 1 wird im linken Teilbaum am ersten freien Platz eingefügt, also als linkes Kind von 3.
- 7 kommt in den rechten Teilbaum, als linkes Kind von 8

- Schrittweiser Aufbau eines Suchbaumes
- Einfügen von Elementen 6, 3, 5, 8, 1, 7, 9, 4:



- Das erste Element "6" wird als Wurzelknoten in den Baum aufgenommen.
- Da $3 < 6$ kommt 3 als linker Nachfolger ("Kind") der Wurzel.
- $5 < 6$ also kommt 5 in den linken Teilbaum. Dort befindet sich ein Knoten mit Inhalt 3. Da $5 > 3$ wird 5 als rechtes Kind von 3 in den Baum eingefügt.
- $8 > 6$, das rechte Kind von der Wurzel existiert noch nicht, also kommt 8 dort hin!
- 1 wird im linken Teilbaum am ersten freien Platz eingefügt, also als linkes Kind von 3.
- 7 kommt in den rechten Teilbaum, als linkes Kind von 8
- 9 wird als rechtes Kind von 8 eingefügt.

- Schrittweiser Aufbau eines Suchbaumes
- Einfügen von Elementen 6, 3, 5, 8, 1, 7, 9, 4:



- Das erste Element "6" wird als Wurzelknoten in den Baum aufgenommen.
- Da $3 < 6$ kommt 3 als linker Nachfolger ("Kind") der Wurzel.
- $5 < 6$ also kommt 5 in den linken Teilbaum. Dort befindet sich ein Knoten mit Inhalt 3. Da $5 > 3$ wird 5 als rechtes Kind von 3 in den Baum eingefügt.
- $8 > 6$, das rechte Kind von der Wurzel existiert noch nicht, also kommt 8 dort hin!
- 1 wird im linken Teilbaum am ersten freien Platz eingefügt, also als linkes Kind von 3.
- 7 kommt in den rechten Teilbaum, als linkes Kind von 8
- 9 wird als rechtes Kind von 8 eingefügt.
- 4 wird in einer neuen Ebene als linkes Kind von 5 eingefügt.

Balancierte Suchbäume

Balancierte Suchbäume sind spezielle Baumstrukturen, die eine effiziente Suche, Einfügen und Löschen von Elementen gewährleisten, indem sie die Höhe des Baumes minimieren. Zu den bekanntesten balancierten Suchbäumen gehören AVL-Bäume und Rot-Schwarz-Bäume.

- **AVL-Bäume:** Diese Bäume sind nach ihrem Erfinder Georgy Adelson-Velsky und Evgenii Landis benannt. Sie garantieren, dass für jeden Knoten die Höhen der linken und rechten Teilbäume sich um höchstens 1 unterscheiden. Dies wird durch Rotationen beim Einfügen und Löschen von Knoten sichergestellt.
- **Rot-Schwarz-Bäume:** Diese Bäume sind eine spezielle Art von binären Suchbäumen, die zusätzlich zu den Suchbaum-Eigenschaften auch Farbregeln einhalten. Jeder Knoten ist entweder rot oder schwarz, und es gelten bestimmte Regeln, die sicherstellen, dass der Baum balanciert bleibt.

Balancierte Suchbäume

Balance-Faktor und Invariante

Für jeden Knoten v sei h_L die Höhe seines linken und h_R die Höhe seines rechten Teilbaums (-1 für leeren Baum). Der *Balance-Faktor* ist

$$bf(v) = h_L - h_R.$$

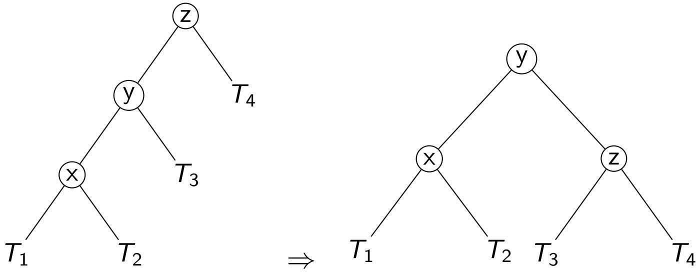
Ein binärer Suchbaum ist ein *AVL-Baum*, wenn für alle Knoten $|bf(v)| \leq 1$ gilt. Dadurch bleibt die Höhe bei n Knoten stets $O(\log n)$;

Rotationen

Verletzt nach Einfügen oder Löschen ein Knoten z (der erste auf dem Rückweg zur Wurzel mit $|bf| > 1$) die Bedingung, betrachten wir dessen schwereren Kindpfad und unterscheiden vier Fälle. X bezeichnet jeweils den neu eingefügten (oder „ursächlich verschobenen") Knoten.

Balancierte Suchbäume

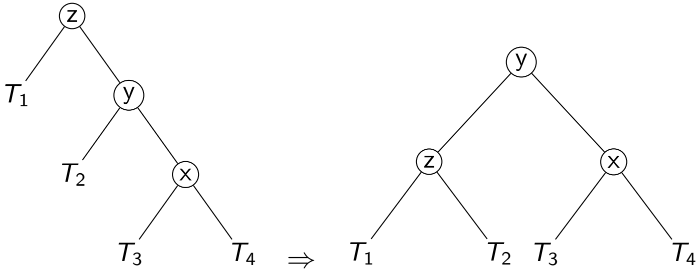
LL-Fall (einfache Rechtsrotation)



LL-Ungleichgewicht ($bf(z) = +2, bf(y) = +1/0$) wird durch Rechtsrotation um z behoben. In einem ausbalancierten Endzustand kann die Höhe von T_3 (linker Teilbaum von z) größer oder gleich der von T_2 sein; dies ist durch die leicht höhere Platzierung angedeutet. Die Inorder-Reihenfolge $T_1 < x < T_2 < y < T_3 < z < T_4$ bleibt erhalten.

Balancierte Suchbäume

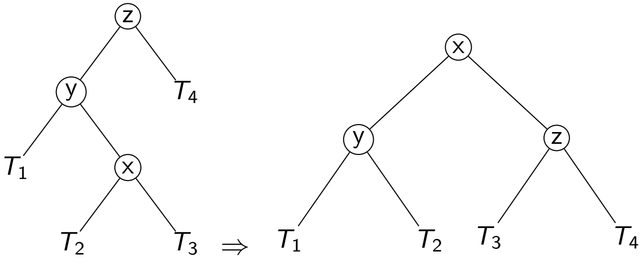
RR-Fall (einfache Linksrotation)



RR-Ungleichgewicht ($bf(z) = -2, bf(y) = -1/0$) wird durch Linksrotation um z behoben. Analog zum LL-Fall kann im Endzustand $h(T_2) \geq h(T_3)$ sein. Die Inorder-Reihenfolge $T_1 < z < T_2 < y < T_3 < x < T_4$ bleibt erhalten.

Balancierte Suchbäume

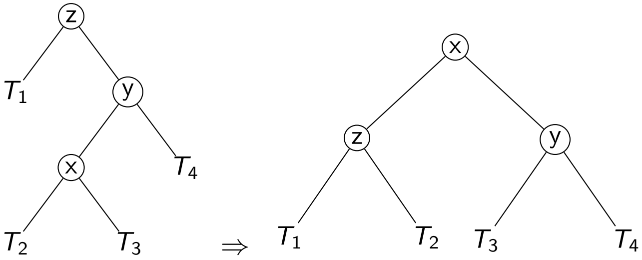
LR-Fall (Doppelrotation: Links dann Rechts)



LR-Ungleichgewicht: Doppelrotation (Linksrotation am linken Kind, dann Rechtsrotation).

Balancierte Suchbäume

RL-Fall (Doppelrotation: Rechts dann Links)



RL-Ungleichgewicht: Doppelrotation (Rechtsrotation am rechten Kind, dann Linksrotation).