Graphentheoretische Konzepte und Algorithmen

Thema 04
Bäume, Wälder und Gerüste

Julia Padberg

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Defintion

Definition (Baum, Wald, azyklischer Digraph)

1. Ein ungerichteter zusammenhängender kreisloser Graph heißt ein *Baum.*
2. Ein ungerichteter Graph, dessen Komponenten Bäume sind, heißt ein *Wald*.
3. Ein gerichteter Graph heißt ein Baum [Wald], wenn der zugrundeliegende ungerichtete Graph ein Baum [Wald] ist.
4. Ein gerichteter kreisloser Graph heißt **azyklisch**[1].
5. Ein Knoten $v \in V$ mit Kontengrad $d(v) = 1$ (bzw. $d_{in}(v) = 1$ und $d_{out}(v) = 0$) in einem Graphen heißt **Endknoten** oder **Blatt** von .

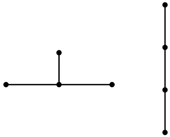
[1] Mitunter wird auch die Bezeichnung DAG als Abkürzung des englischen „directed acyclic graph" verwendet.

Beispiel

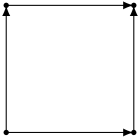
BSP:

In der folgenden Abbildung ist G_1 ein Wald und G_2 ein azyklischer Digraph:

G_1 :



G_2 :



Die hier gewählt Darstellung betont den Umstand, daß i.allg. keine spezielle Art der visuellen Darstellung mit Bäumen verbunden wird.

Aufgabe 1:

Zeigen Sie bitte für ungerichtete Graphen:
G ist ein Baum, gdw. es zwischen je zwei verschiedenen Knoten aus G genau einen Weg gibt.

Lösung

$\Rightarrow$ Indirekter Beweis:
Sei kein Weg von dem Knoten u zum Knoten v gegeben, dann ist G nicht zusammenhängende, also kein Baum.
Seien mindestens zwei Wege von dem Knoten u zum Knoten v gegeben und w die Weggabelung und sei x das Wiederzusammentreffen, dann bilden die beiden Teilstrecken der Wege zwischen w und x bilden einen Kreis, also ist G kein Baum.

$\Leftarrow$ Es gibt genau einen Weg, also ist der Graph zusammenhängend und kreislos, also ein Baum.

Baumwachstum

Satz

Sei G ein Graph, der einen Endknoten v enthält. Dann sind folgende Aussagen äquivalent:

i. G ist ein Baum.

ii. G/v ist ein Baum.

Beweis: $(i) \Rightarrow (ii)$: Betrachte zwei Knoten w, w' in G. Da G zusammenhängt, sind w und w' durch einen Pfad verbunden. Dieser Pfad enthält keinen Knoten mit Grad 1 (außer evtl. w, w'). Also enthält er nicht v. Also ist der Pfad vollständig in G/v, und ist somit zusammenhängend. Da G keinen Kreis enthält, kann G/v auch keinen Kreis enthalten. Damit ist G/v ein Baum.

$(ii) \Rightarrow (i)$: Durch Hinzufügen eines Endknotens zu G/v kann kein Kreis entstehen. Es gibt einen Pfad von v über den (eindeutigen) Nachbarn v' von zu jedem anderen Knoten in G. Also ist G ein Baum.

Eigenschaften

Satz

Sei $G = (V, E)$ ein ungerichteter Graph. Dann sind die folgenden Aussagen äquivalent:

1. G ist ein Baum.
2. Zwischen je zwei verschiedenen Knoten aus G gibt es genau einen Weg.
3. G ist zusammenhängend, und jede Kante aus E ist eine Schnittkante.
4. G ist zusammenhängend, und es ist $|E| = |V| - 1$.
5. G besitzt keinen Kreis, aber durch Hinzufügen einer beliebigen Kante zu E entsteht ein Graph mit genau einem Kreis.

Beweisidee: Ringschluss $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4 \Rightarrow 5 \Rightarrow 1$

$1 \Rightarrow 2$: Gegeben seien zwei Wege von u nach v.
Sei w die Gabelung und x der Knoten das Wiederzusammentreffen.
Die beiden Teilstrecken der Wege zwischen w und x bilden einen Kreis: $\lightning$

$2 \Rightarrow 3$: G ist zusammenhängend, denn je zwei Knoten sind miteinander verbunden. Da es nach Voraussetzung nur einen Weg zwischen zwei Knoten gibt, zerstört die Wegnahme einer Kante den Zusammenhang zwischen ihren Endknoten; sie ist also eine Schnittkante.

$3 \Rightarrow 4$: Wenn $d(v) > 1$ für alle Knoten, dann gibt es einen Kreis. Eine Kante auf einem Kreis ist keine Schnittkante, also muss ein Graph mit 3. einen Knoten vom Grad 1 besitzen oder K_1 sein. Wiederholte Wegnahme dieses Knoten samt der inzidenten Kante ändert nichts. Zum Schluss bleibt K_1. Also genausoviele Knoten wie Kanten gelöscht und ein Knoten übrig.

Beweisidee: Ringschluss $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4 \Rightarrow 5 \Rightarrow 1$

$4 \Rightarrow 5$: Wenn $|V| - 1$ Kanten, dann gibt es mindestens einen Knoten mit $d(v) = 1$. Nach Löschen dieses Knoten samt der inzidenten Kante immer noch $|V| - 1$ Kanten, also wiederholen, bis der Graph zum K_1 geworden ist. Kreis enhält aber keinen Knoten mit $d(v) = 1$, also kreisfrei.
Durch Hinzufügen einer neuen Kante entsteht zwischen ihren inzidenten Knoten ein neuer, alternativer Weg und damit insgesamt ein Kreis.
Es ist genau ein Kreis, denn sonst wäre G vorher nicht kreisfrei gewesen.

$5 \Rightarrow 1$: G muß zusammenhängend sein, denn sonst würde durch das Einfügen einer Kante zwischen zwei Komponenten kein Kreis entstehen.

Wurzelbaum

Definition (Wurzel,Wurzelbaum)

1. Sei $G(V, E)$ ein ungerichteter Baum. Um einen Knoten von G in besonderer Weise hervorzuheben, kann sie als **Wurzel** von G bezeichnet werden. G heißt dann ein Wurzelbaum. Die Knoten vom Grad 1 heißen *Blätter* des Wurzelbaums.
2. Sei $G = (V, E)$ ein gerichteter Baum. Der Knoten $v \in V$ heißt *Wurzel von* G, wenn alle anderen Knoten in V von v aus erreichbar sind. Gilt für ein $v \in V : d_{out}(v) = 0 \wedge d_{in}(v) > 0$, so heißt v *Blatt*.

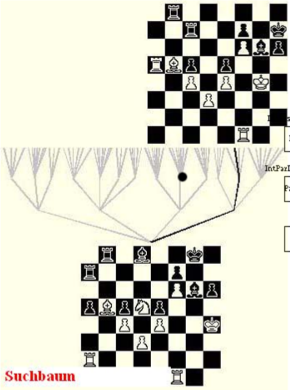
Bemerkung: Aus der Def. folgt direkt, dass für Blätter im gerichteten Wurzelbaum gilt: $d_{in}(v) = 1$.

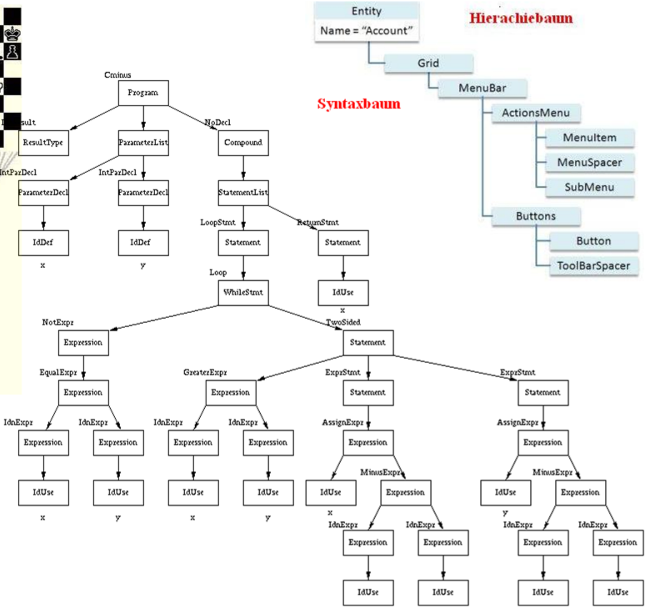
Wurzelbaum

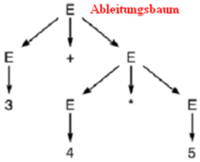
Satz

In einem Wurzelbaum B bildet jeder Knoten zusammen mit ihren Nachkommen und den dazugehörigen Kanten wieder einen Wurzelbaum.

Beispiele


Suchbaum


Syntaxbaum
Cminus
Program
ResultType
ParameterList
NoDecl
Compound
IntParDecl
IntParDecl
ParameterDecl
ParameterDecl
StatementList
IdDef
IdDef
x
y
LoopStmt
ReturnStmt
Statement
Statement
Loop
WhileStmt
IdUse
x
NotExpr
TwoSided
Expression
Statement
EqualExpr
GreaterExpr
ExprStmt
ExprStmt
Expression
Expression
Statement
Statement
IdnExpr
IdnExpr
IdnExpr
IdnExpr
AssignExpr
AssignExpr
Expression
Expression
Expression
Expression
Expression
Expression
MinusExpr
MinusExpr
IdUse
IdUse
IdUse
IdUse
IdUse
Expression
IdUse
Expression
x
y
x
y
x
y
IdnExpr
IdnExpr
IdnExpr
IdnExpr
Expression
Expression
Expression
Expression
IdUse
IdUse
IdUse
IdUse
Hierachiebaum
Entity
Name = "Account"
Grid
MenuBar
ActionsMenu
MenuItem
MenuSpacer
SubMenu
Buttons
Button
ToolBarSpacer


Ableitungsbaum
E
E
+
E
3
E
*
E
4
5

Binäre Bäume

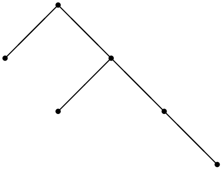
Definition (Binäre Bäume)

Ein Wurzelbaum heißt *binär*, wenn die Wurzel einen Grad ≤ 2 besitzt und alle anderen Knoten mit Ausnahme der Blätter vom Grad 2 oder 3 sind, also maximal zwei Kinder haben. Ein binärer Baum heißt *balanciert*, wenn zwischen jedem Blatt und der Wurzel dieselbe Anzahl von Kanten liegt und mit Ausnahme der Blätter jeder Knoten genau zwei Kinder besitzt. Von diesen wird einer als *rechtes Kind* und der andere als *linkes Kind* bezeichnet.

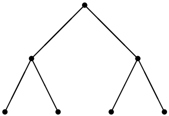
Hinweis:

Im Bereich der Algorithmen und Datenstrukturen wird oft eine abgeschwächte Version von *balanciert* verwendet. Im extremen Fall wird nur verlangt, daß alle Blätter zur Wurzel die gleiche Anzahl an Kanten haben.

Beispiele



binärer Baum



balancierter binärer Baum

Definition

Der Abstand $a(x, y)$ von zwei Knoten eines Baumes ist die Länge des (eindeutigen) Weges zwischen ihnen.

.... also die Anzahl der Kanten auf diesem Weg

Länge und Höhe eines Wurzelbaumes

Definition

- Sei B ein Wurzelbaum. Der Maximalwert unter den Abständen $a(x, x_0)$ der Knoten $x \in B$ zur Wurzel x_0 heisst Länge L des Wurzelbaumes.
- Die Anzahl der Knoten des längsten Weges zur Wurzel x_0 heißt Höhe H des Wurzelbaumes.

. . . . also Höhe= Länge+1
(Deswegen braucht man nur eine der beiden Definitionen.)

Satz

Sei B ein Binärbaum mit Höhe H.
Dann enthält B maximial $2^H - 1$ Knoten.

Aufgabe 2: Beweis durch Induktion !!!

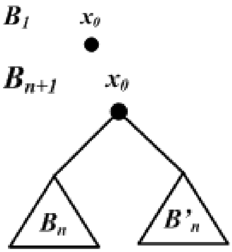
Lösung von Aufgabe 2

Beweis durch Induktion über H:

IA $H = 1$, dann ist der maximale Binärbaum B_1 mit $1 = 2^1 - 1 = 2^H - 1$ Knoten

IB Ein Binärbaum mit Höhe H enthält maximial $2^H - 1$ Knoten.

IS Sei B_{n+1} ein maximaler Binärbaum mit Höhe $H + 1$, dann besteht er aus dem Wurzelknoten x_0 und den beiden maximalen binären Bäumen B_n und B'_n. B_n und B'_n haben die Höhe H und damit nach IB $2^H - 1$ Knoten.
B_{n+1} hat also
$2 \cdot (2^H - 1) + 1 = 2 \cdot 2^H - 2 + 1 = 2^{H+1} - 1$ Knoten.



Suchbäume

Um in einem Datenbestand mit 100.000 Einträgen jeden Eintrag schnell wiederzufinden, ist ein Suchbaum die richtige Datenstruktur. Dafür müssen Ihre Daten einen sortierbaren Schlüssel besitzen.
Ein Suchbaum entsteht durch folgenden Algorithmus:

Algorithmus

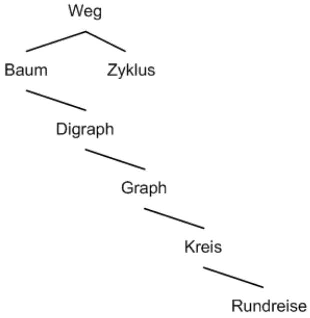
Trage Schlüssel s in den Baum ein:
> Existiert Baum noch nicht, so erzeuge Wurzel und trage s als Wurzelschlüssel ein.
> Sei s_0 der Wurzelschlüssel.
> Sonst:
>> Falls $s < s_0$: Trage Schlüssel s im linken Teilbaum der Wurzel ein.
>> Falls $s > s_0$: Trage Schlüssel s im rechten Teilbaum der Wurzel ein.

Aufgabe 3:

Erstellen Sie bitte einen Suchbaum. Fügen Sie in der folgenden, vorgegebenen Reihenfolge die Elemente in einen leeren Such-Baum ein. Als Ordnungsrelation gilt die lexikographische Ordnung.

Weg, Baum, Digraph, Graph, Kreis, Rundreise, Zyklus

Lösung



Suche im Suchbaum

Die Suche eines bestimmten Eintrages geschieht durch folgende rekursive Suchvorschrift:

Algorithmus

Suche s beginnend bei :
Ist s gleich dem Schlüssel x, so liefere x zurück.
Sonst:
Falls $s < x$: Suche s beginnend beim linken Sohn von x
Falls $s > x$: Suche s beginnend beim rechten Sohn von x

Aufgabe 4: : Wieviele Knoten kann man maximal speichern, wenn man bis zu 19 Vergleiche erlaubt?

Lösung der Aufgabe 4

Lösung

Mit der Suchstrategie „Suche s beginnend bei x" braucht man in einem Baum der Höhe H maximal $H - 1$ Vergleiche, um alle Knoten, also $2^H - 1$ zu finden.
Bei 19 Vergleichen sind also maximal $2^{18} - 1 \sim 262.000$ Einträge speicherbar.

AVL-Bäume

- AVL-Bäume sind nach ihren Begründern **A**delson, **V**elskii und **L**andis (1962) benannt
- Algorithmus zum Aufbau eines „balancierten“ Suchbaums
- Ein beliebiger Suchbaum kann nur in einen AVL-Baum transformiert werden, indem alle Elemente neu eingefügt werden.
- Nur sinnvoll, wenn sehr viele lesende Zugriffe nach einer aufwendigen Aufbauphase
- Löschen ist auch möglich in AVL-Bäumen.
- z.B. von Linux teilweise für die Verwaltung von Speicherplätzen verwendet!

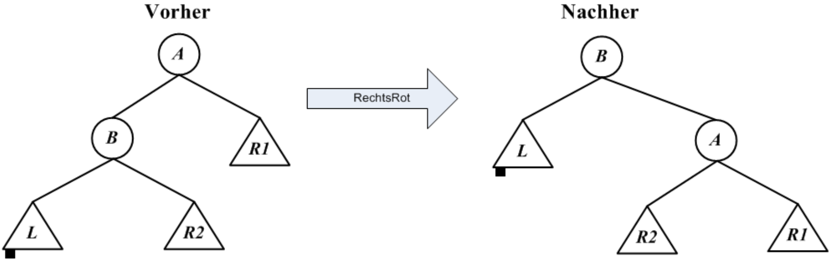
Vorgehen bei der Konstruktion von AVL-Bäumen

- neues Element in den binären Suchbaum einfügen wie bisher
  - wesentlich ist die Balance
    Balance = Höhe rechter Teilbaum - Höhe linker Teilbaum
  - zulässige Werte sind nur -1, 0, 1
  - sonst Neubalancierung durch Rotation
- nach dem Einfügen eines Elementes *bottom up* überprüfen, ob eine der vier Situationen
  - Linksrotation
  - Rechtsrotation
  - Linksproblem
  - Rechtsproblem

Rechtsrotation

Bedingung: Die Höhe des Teilbaums **R1** ist um 2 niedriger, als die Höhe des Teilbaums mit Wurzel **B**. Der Unterschied sei durch den Teilbaum **L** begründet.

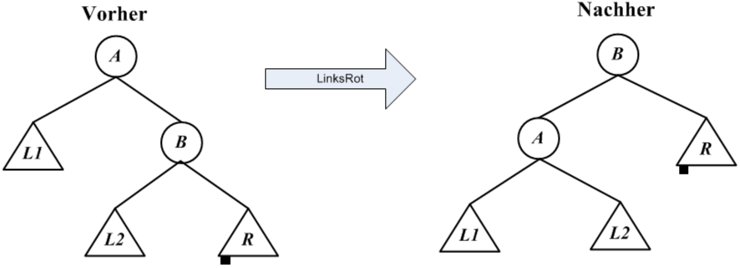
- **Vorher:** Die Höhe von **R1** sei **n**. Dann ist die Höhe von **R2** gleich **n** oder **n - 1** und die Höhe von **L** gleich **n + 1**. Die Gesamthöhe ist **n + 3**.
- **Nachher:** Die Höhe von dem Teilbaum mit Wurzel **A** ist **n + 1**, also genauso wie die Höhe von **L**! Die Gesamthöhe ist **n + 2**.



Linksrotation

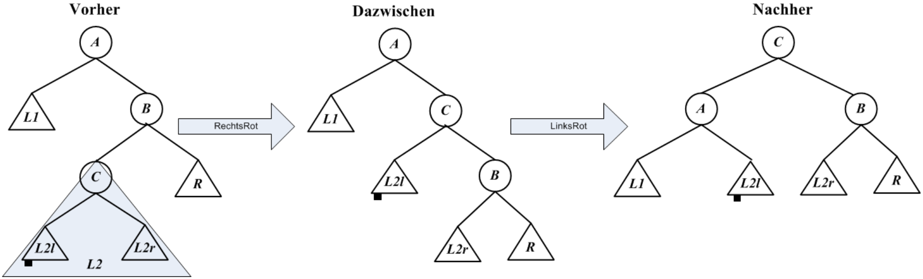
Bedingung: Die Höhe des Teilbaums `L1` ist um 2 niedriger, als die Höhe des Teilbaums mit Wurzel `B`. Der Unterschied sei durch den Teilbaum `R` begründet.

- **Vorher**: Die Höhe von `L1` sei `n`. Dann ist die Höhe von `L2` gleich `n` oder `n – 1` und die Höhe von `R` gleich `n + 1`. Die Gesamthöhe ist `n + 3`.
- **Nachher**: Die Höhe von dem Teilbaum mit Wurzel `A` ist `n + 1`, also genauso wie die Höhe von `R`! Die Gesamthöhe ist `n + 2`.



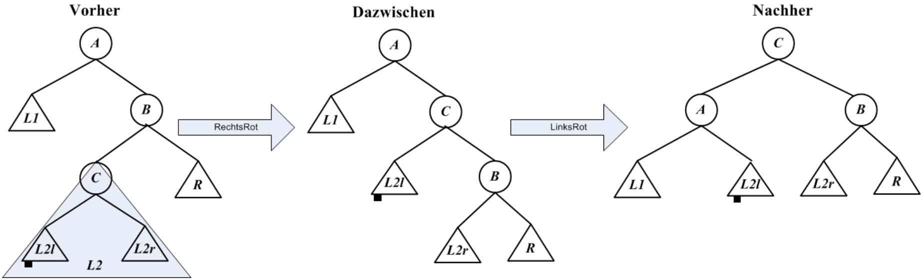
Problemsituation links

Bedingung: Die Höhe des Teilbaums `L1` ist um 2 niedriger, als die Höhe des Teilbaums mit Wurzel `B`. Der Unterschied sei durch den Teilbaum `L2` begründet. Dieser Teilbaum ist in der Abbildung detaillierter mit Wurzel `C` und den beiden Teilbäumen `L2l` und `L2r` dargestellt. In diesem Fall ist zuerst in dem Teilbaum mit Wurzel `B` eine Rechtsrotation durchzuführen (siehe in der Abbildung die Darstellung `Dazwischen`) und dann in dem Baum mit Wurzel `A` eine Linksrotation durchzuführen.



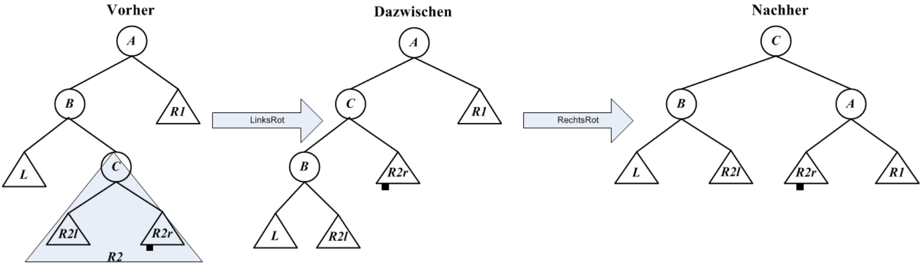
Problemsituation links (ff)

- **Vorher:** Die Höhe von `L1` sei `n`. Dann ist die Höhe von `R` gleich `n` oder `n - 1` und die Höhe von `L2` (Teilbaum mit Wurzel `C`) gleich `n + 1`. In der dargestellten Situation ist die Höhe von `L2l` `n`; die Höhe von `L2r` ist dann `n - 1`.
  Die Gesamthöhe ist `n + 3`.
- **Nachher:** Die Höhe von dem Teilbaum mit Wurzel `A` ist `n + 1` und die Höhe von dem Teilbaum mit Wurzel `B` ist `n` oder `n + 1`. Die Gesamthöhe ist `n + 2`.



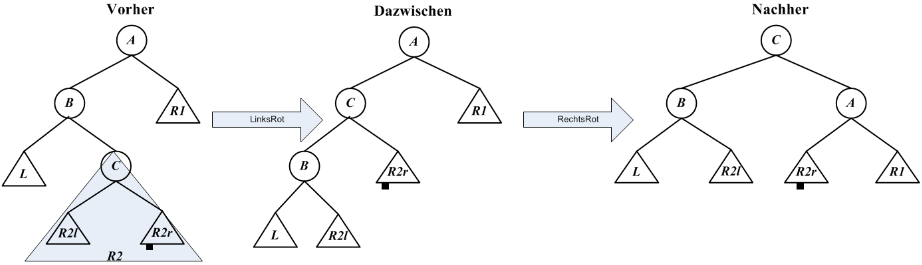
Problemsituation rechts

Bedingung: Die Höhe des Teilbaums `R1` ist um 2 niedriger, als die Höhe des Teilbaums mit Wurzel `B`. Der Unterschied sei durch den Teilbaum `R2` begründet. Dieser Teilbaum ist in der Abbildung detaillierter mit Wurzel `C` und den beiden Teilbäumen `R2l` und `R2r` dargestellt. In diesem Fall ist zuerst in dem Teilbaum mit Wurzel `B` eine Linksrotation durchzuführen (siehe in der Abbildung die Darstellung `Dazwischen`) und dann in dem Baum mit Wurzel `A` eine Rechtssrotation durchzuführen.



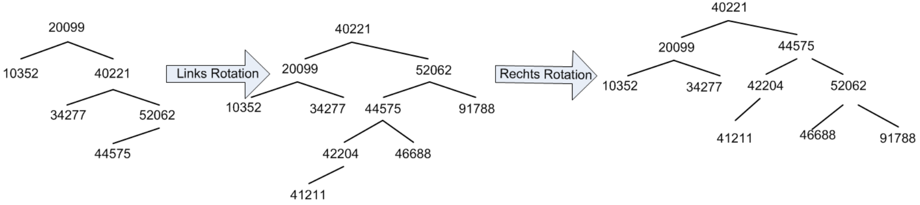
Problemsituation rechts (ff)

- **Vorher:** Die Höhe von `R1` sei `n`. Dann ist die Höhe von `L` gleich `n` oder `n - 1` und die Höhe von `R2` (Teilbaum mit Wurzel `C`) gleich `n + 1`. In der dargestellten Situation ist die Höhe von `R2r` `n`; die Höhe von `R2l` ist dann `n - 1`.
  Die Gesamthöhe ist `n + 3`.
- **Nachher:** Die Höhe von dem Teilbaum mit Wurzel `A` ist `n + 1` und die Höhe von dem Teilbaum mit Wurzel `B` ist `n` oder `n + 1`. Die Gesamthöhe ist `n + 2`.



Beispiel

Die folgenden Postleitzahlen der Reihe nach in AVL-Baum einsortieren: **20099, 10352, 40221, 52062, 34277, 44575, 91788, 46688, 42204, 41211**

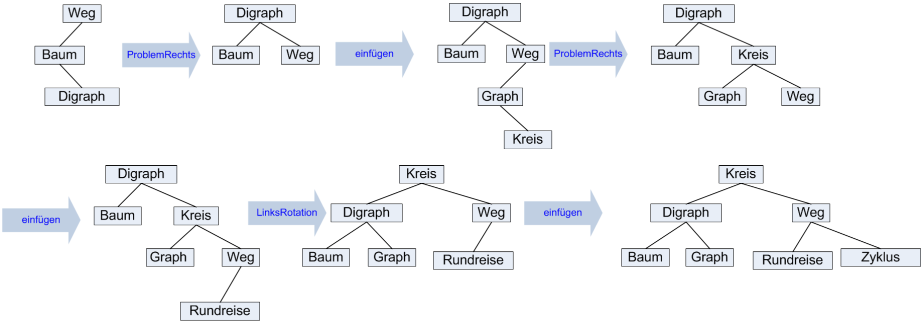


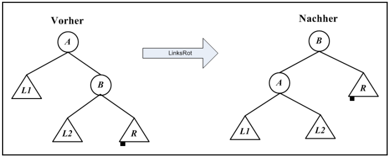
Aufgabe 5:

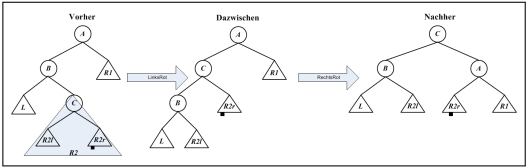
Erstellen Sie bitte einen AVL-Baum. Fügen Sie in der folgenden, vorgegebenen Reihenfolge die Elemente in einen leeren AVL-Baum ein.
Als Ordnungsrelation gilt die lexikographische Ordnung.
Erläutern Sie, wann und welche Rotationen stattfinden und zeichnen Sie den Baum nach jeder Rotation neu auf.

Weg, Baum, Digraph, Graph, Kreis, Rundreise, Zyklus

Lösung von Aufgabe 5



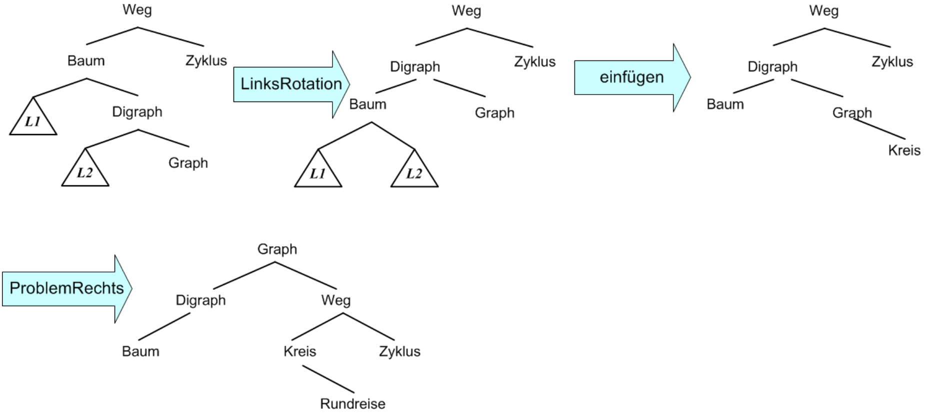




Aufgabe 6:

Erstellen Sie bitte einen AVL-Baum. Fügen Sie in der folgenden, vorgegebenen Reihenfolge die Elemente in einen leeren AVL-Baum ein.
Als Ordnungsrelation gilt die lexikographische Ordnung.
Erläutern Sie, wann und welche Rotationen stattfinden und zeichnen Sie den Baum nach jeder Rotation neu auf.

Weg, **Zyklus**, Baum, Digraph, Graph, Kreis, Rundreise


Weg
Baum
Zyklus
L1
Digraph
L2
Graph
LinksRotation
Weg
Digraph
Zyklus
Baum
Graph
L1
L2
einfügen
Weg
Digraph
Zyklus
Baum
Graph
Kreis
ProblemRechts
Graph
Digraph
Weg
Baum
Kreis
Zyklus
Rundreise

Gerüst

Definition

Sei $G = (V, E)$ ein ungerichteter Graph. Ein Baum $H = (W, F)$ heißt ein **Gerüst** (auch Spannbaum, aufspannender Baum) von G, wenn H ein Teilgraph von G ist und alle Knoten von G enthält (wenn also gilt $F \subseteq E$ und $W = V$).

Wenn H ein Gerüst von G ist, sagt man auch: „G wird von H aufgespannt“.

Zusammenhang und Gerüste

Satz

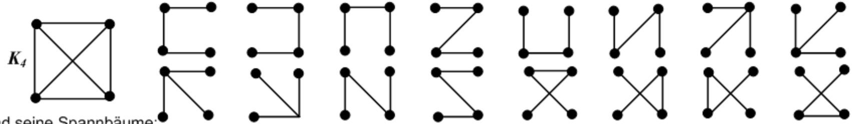
Ein Graph G besitzt ein Gerüst, gdw. er zusammenhängend ist.

Beweis: $\Rightarrow$ trivial
$\Leftarrow$ sei $G = (V, E)$ nun zusammenhängend.

- 1. Fall $|V| = |E| + 1$ Dann ist G das gesuchte Gerüst selbst (Folie 150).
- 2. Fall $|V| > |E| + 1$ Dann kann nicht zusammenhängend sein, Widerspruch.
- 3. Fall $|V| < |E| + 1$ Dann ist G kein Baum, und muss mindestens einen Kreis enthalten. Durch das Entfernen einer beliebige Kante e des Kreises, ergibt sich wieder ein zusammenhängender Graph $G' = (V, E')$ mit $E' = E \setminus \{e\}$. Entweder gilt nun $|V| = |E'| + 1$ und G' ist das gesuchte Gerüst. Oder wir wiederholen diese Reduktion, was möglich ist, da G' zusammenhängend ist.

Siehe auch YouTube: Graphen & Algorithmen: Kapitel 2 - Bäume (5)

Beispiel

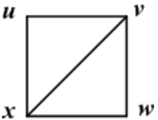


Und seine Spannbäume:

Naheliegende Fragen:

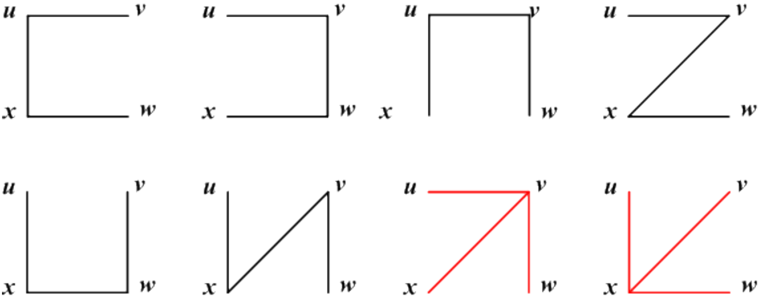
- Hat jeder Graph einen aufspannenden Baum?
- Wie findet man einen solchen Baum, so es ihn gibt?
- Gibt es eine kleinsten aufspannenden Baum?

Aufgabe 7:



besitzt 8 Gerüste.
Welche?
Wieviele davon sind nichtisomorph?

Lösung



Anzahl der Gerüste

Definition

Gegeben ein ungerichteter Graph G, dann ist $\tau(G)$ die Anzahl der verschiedenen Gerüste von G.

Satz

Satz von Caley Der vollständige Graph K_n mit n Knoten hat $\tau(K_n) = n^{n-2}$ verschiedene (nicht notwendig nichtisomorphe) Gerüste.

Bemerkung

Der Beweis macht sich den Prüfer-Code, das ist ein eindeutige Kodierung von binären Bäumen mit n Knoten durch ein Tupel mit $n - 2$ Zahlen zwischen 1 und n, zunutze.

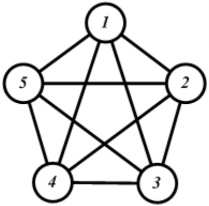
Beweisidee

1. Gerüste werden Tupel zugeordnet (Prüfer-Code)
2. Anzahl der Tupel = Anzahl der Gerüste
3. Konstruktion:
   3.1 Tupel T gibt es genau ein S_T
   3.2 Nachweis, dass S_T Gerüst ist
4. für jedes Gerüst S gibt es genau ein Tupel T_S

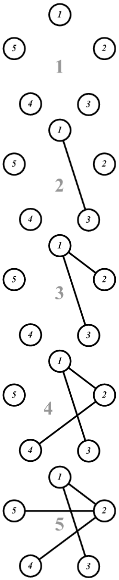
Konstruktion

- $|V| = n$, dann aus $n-2$-Tupel T wird Spannbaum $S = \{1, 2, ..., n\}$ konstruiert.
- X_i noch zu verbindende Knoten
- Rest des Tupels $T_i = (j_i, j_{i+1}, ..., j_{n-2})$
- k_i das Minimum der Zahlen aus X, die nicht in T_i vorkommen
- neue Kante $e_i = (k_i, j_i)$ also $n-2$ Kanten
- letzte Kante X_{n-1}

BSP: Tupel $\Longrightarrow$ Gerüst



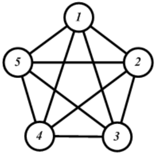
Gegeben K_5:



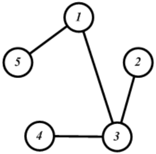
1. Gegeben sei als BSP $T = (1, 2, 2) = T_1$
2. Dann $X_1 = V = \{1, 2, 3, 4, 5\}$ und $k_1 = min(X_1 \setminus T_1) = 3$ und $e_1 = (3, 1)$
3. Dann $X_2 = \{1, 2, 4, 5\}$ und $k_2 = min(X_2 \setminus T_2) = 1$ und $e_2 = (1, 2)$
4. Dann $X_3 = \{2, 4, 5\}$ und $k_3 = min(X_3 \setminus T_3) = 4$ und $e_3 = (4, 2)$
5. Dann $X_4 = \{2, 5\}$ und $e_4 = (2, 5)$

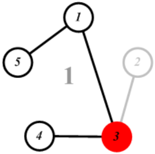
BSP: Gerüst $\Longrightarrow$ Tupel

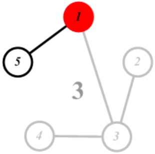
Gegeben K_5:

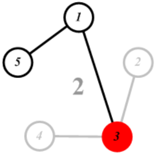


Gegeben dieses Gerüst:









1. $T = (3, \_, \_)$
2. $T = (3, 3, \_)$
3. $T = (3, 3, 1)$

Beweis (1)

Gegeben sei K_n mit $V = \{1, 2, ..., n\}$.

1. Dann ist zu zeigen, dass das $n - 2$-Tupel $T = (j_1, j_2, ..., j_{n-2})$ mit Werten $1 \leq j_i \leq n$ für $1 \leq i \leq n$ genau ein Gerüst beschreibt.
2. Es gibt n^{n-2} solche Tupel, kombinatorisch Auswahl von $n - 2$ aus n Elementen mit Wiederholung.

Beweis (2)

3. Konstruktion:

3.1 Sei T ein beliebiges Tupel mit $T = T_1 = (j_1, j_2, ..., j_{n-2})$ mit Werten $1 \leq j_i \leq n$. Dann wird S konstruiert, so dass S alle Knoten $\{1, 2, ..., n\}$ hat :

- $X_1 := V$,
  $k_1 := min(X_1 \setminus T_1)$
  das Minimum der Zahlen aus X, die nicht in T_1 vorkommen $e_1 = (k_1, j_1)$ erste Kante
- Für $2 \leq i \leq n-2$:
  $X_i := X_{i-1} \setminus \{k_{i-1}\}$
  $T_i = (j_i, j_{i+1}, ..., j_{n-2})$ $k_i := min(X_i \setminus T_i)$
  das Minimum der Zahlen aus X, die nicht in T_i vorkommen $e_i = (k_i, j_i)$ weitere Kanten
- $X_{n-1} = \{k_{n-1}, k_n\}$
  $e_{n-1} = (k_{n-1}, k_n)$ letzte Kante

Für jedes T gibt es genau ein S_T.

Beweis (3)

3. Konstruktion:

   3.2 S_T ist Gerüst:
   S_T hat $n-1$ Kanten laut Konstruktion.
   zu zeigen: S_T ist kreisfrei:
   Es gilt (für die implizite Richtung):

   a) Bei der Konstruktion werden die Knoten aus X_i gelöscht, also ist jeder Knoten ist Anfangspunkt höchstens einer Kante und k_n ist Anfangspunkt von keiner Kante.
   b) Weil die X_i keine Knoten des Tupels enthalten, ist kein Endpunkt einer Kante Anfangspunkt einer anderen Kante.

   Annahme: S_T enthalte einen Kreis, dann gilt

   - entweder haben alle Kanten die gleiche implizite Richtung, also gibt es Knoten die Anfangspunkt und Endpunkt sind,
     dann Widerspruch wegen b)
   - oder die Kanten haben unterschiedliche implizite Richtung, also gibt es Knoten die Anfangspunkt zweier Kanten sind,
     dann Widerspruch wegen a)

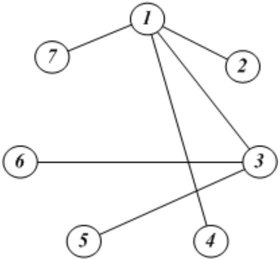
Beweis (4)

1. Für jedes Gerüst gibt es genau ein Tupel:
   Gegeben sei Gerüst $S = S_1$, dann wird T_S wie folgt erzeugt:
   1.1 j_i ist einziger Nachbar des kleinsten Knoten mit Knotengrad 1 in S_i
   1.2 $S_{i+1} = S_i \setminus i$, also S_{i+1} ist S_i ohne den kleinsten Knoten mit Knotengrad 1 in S_i

   Diese Konstruktion ergibt für jedes Gerüst genau ein Tupel.

Aufgabe 8:

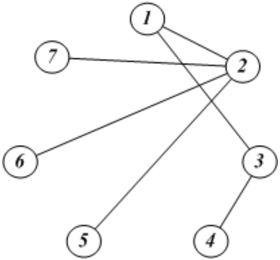
Berechnen Sie bitte für K_7, dann Spannbaum, der durch $(1, 1, 3, 3, 1)$ kodiert wird.



- $min(X_1 \setminus T_1) = min(\{1, ..., 7\} \setminus \{1, 3\}) = 2$ also $1 - 2$
- $min(X_2 \setminus T_2) = min(\{1, 3, ..., 7\} \setminus \{1, 3\}) = 4$ also $1 - 4$
- $min(X_3 \setminus T_3) = min(\{1, 3, 5, 6, 7\} \setminus \{1, 3\}) = 5$ also $3 - 5$
- $min(X_4 \setminus T_4) = min(\{1, 3, 6, 7\} \setminus \{1, 3\}) = 6$ also $3 - 6$
- $min(X_5 \setminus T_5) = min(\{1, 3, 7\} \setminus \{1\}) = 3$ also $3 - 1$
- $X_6 = \{1, 7\}$ also $1 - 7$

Aufgabe 9:

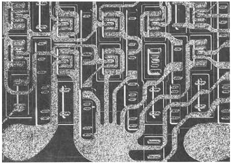
Berechnen Sie bitte für K_7, das Tupel, dass diesen Spannbaum kodiert wird.



$(3, 1, 2, 2, 2)$

Direkte Anwendungen von Minimalgerüsten

- Entwurf physikalischer Netz-Systeme, bei denen Ausfallsicherheit keine Rolle spielt
- Beispiele:
  - Straßen
  - Stromnetz
  - Wasser-, Gas- oder Ölrohrleitungsnetze
  - Entwurf von Mikrochips
  - Vernetzung von Rechnern



- Knoten: Teilnehmer / Erzeuger / Abnehmer
- Kanten: Mögliche Verbindung zwischen Knoten
- Gewichte: Verbindungskosten

Aufgabenstellung: Minimal-und Maximalgerüste

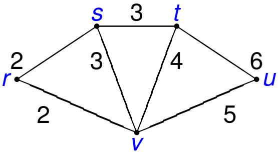
Die wichtigste Aufgabenstellung im Zusammenhang mit Gerüsten verlangt die Bestimmung eines sogenannten Minimalgerüsts [Maximalgerüste].

Aufgabenstellung

Vorgelegt sei ein zusammenhängender schlichter ungerichteter Graph $G(V, E)$, bei dem jede Kante $v_i v_j$ mit einer nichtnegativen Bewertung l_{ij} („Länge“) versehen ist. Gesucht ist ein Gerüst von G mit minimaler [maximaler] Kantenbewertungssumme.

Beispiel

Die Gemeinden eines Kreises haben gemeinsam einen Schneepflug angeschafft. Bei Schneefall soll er dafür sorgen, daß zwischen je zwei Gemeinden eine (nicht notwendigerweise direkte) Straßenverbindung erhalten bleibt. Dabei soll die Gesamtlänge der zu räumenden Straßen möglichst klein sein (damit diese Straßen lieber öfters geräumt werden können).
Das Straßennetz (mit Längenangaben in km) hat folgende Gestalt:



Algorithmus von Kruskal

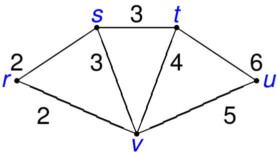
Algorithmus

Eingabe: Eine Menge E der Kanten mit ihren Längen.
Ausgabe: Eine Teilmenge F der Kantenmenge.

- Numeriere die Kanten $e_1, \ldots, e_{|E|}$ nach steigender Länge. Setze $F := \emptyset$.
- Für $i := 1, \ldots, |E|$:
  - Falls $F \cup \{e_i\}$ nicht die Kantenmenge eines Kreises in G enthält, setze $F := F \cup \{e_i\}$.

Beispiel

Für



erhalten wir folgende Sortierung der Kanten:

$$e_1 = rs,\ e_2 = rv,\ e_3 = st,\ e_4 = sv,\ e_5 = tv,\ e_6 = uv,\ e_7 = tu$$

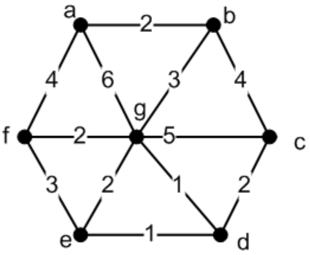
und das (in diesem Fall eindeutig bestimmte) Minimalgerüst enthält die Kanten

$$F = \{rs,\ rv,\ st,\ uv\}$$

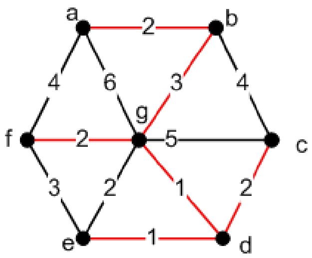
mit der Kantengewichtssumme 12.

Aufgabe 10:

Bestimmen Sie ein Minimalgerüst des Graphen



Lösung von Aufgabe 10



mit
$e_1 = ed$, $e_2 = dg$, $e_3 = dc$, $e_4 = eg$, $e_5 = ab$, $e_6 = fg$, $e_7 = fe$, $e_8 = gb$, $e_9 = af$, $e_{10} = bc$, $e_{11} = gc$, $e_{12} = ag$

dann $F = \{e_1, e_2, e_3, e_5, e_6, e_8\}$