Studientag zur Algorithmischen Mathematik

Eulertouren, 2-Zusammenhang, Bäume und Baumisomorphismen

Winfried Hochstättler

Diskrete Mathematik und Optimierung
FernUniversität in Hagen

22. Mai 2011

Outline

Eulertouren

Eine Eulertour ist ein geschlossener Spaziergang, der jede Kante genau einmal besucht.

Eulertouren

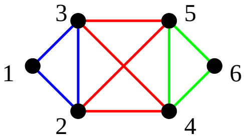
Eine Eulertour ist ein geschlossener Spaziergang, der jede Kante genau einmal besucht.

G ist eulersch $:\iff$ G hat Eulertour

Eulertouren

Eine Eulertour ist ein geschlossener Spaziergang, der jede Kante genau einmal besucht.

G ist eulersch $:\Longleftrightarrow$ G hat Eulertour

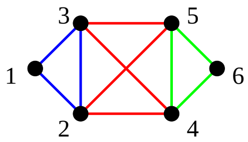


ist eulersch

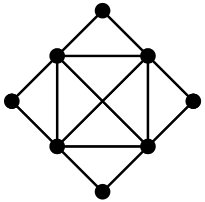
Eulertouren

Eine Eulertour ist ein geschlossener Spaziergang, der jede Kante genau einmal besucht.

G ist eulersch $:\iff$ G hat Eulertour



ist eulersch



ist nicht eulersch

Eulersche Graphen

Satz

Sei $G = (V, E)$ ein Graph. Paarweise äquivalent sind

(i) *G ist eulersch.*

(ii) *G ist zusammenhängend und $\forall v \in V : \deg(v)$ ist gerade.*

(iii) *G ist zusammenhängend und E ist kantendisjunkte Vereinigung von Kreisen.*

Eulersche Graphen

Satz

Sei $G = (V, E)$ ein Graph. Paarweise äquivalent sind

(i) *G ist eulersch.*

(ii) *G ist zusammenhängend und $\forall v \in V : \deg(v)$ ist gerade.*

(iii) *G ist zusammenhängend und E ist kantendisjunkte Vereinigung von Kreisen.*

Beweis.

$(i) \Rightarrow (ii)$: klar (genauso oft rein wie raus).

Eulersche Graphen

Satz

Sei $G = (V, E)$ ein Graph. Paarweise äquivalent sind

(i) *G ist eulersch.*

(ii) *G ist zusammenhängend und* $\forall v \in V : \deg(v)$ *ist gerade.*

(iii) *G ist zusammenhängend und E ist kantendisjunkte Vereinigung von Kreisen.*

Beweis.

$(i) \Rightarrow (ii)$: klar (genauso oft rein wie raus).

$(ii) \Longrightarrow (iii)$: Wähle $v_0 \in V$ und laufe solange noch neue Kanten gefunden werden.

Eulersche Graphen

Satz

Sei $G = (V, E)$ ein Graph. Paarweise äquivalent sind

(i) *G ist eulersch.*

(ii) *G ist zusammenhängend und $\forall v \in V : \deg(v)$ ist gerade.*

(iii) *G ist zusammenhängend und E ist kantendisjunkte Vereinigung von Kreisen.*

Beweis.

$(i) \Rightarrow (ii)$: klar (genauso oft rein wie raus).

$(ii) \Longrightarrow (iii)$: Wähle $v_0 \in V$ und laufe solange noch neue Kanten gefunden werden. Da $\deg(v)$ gerade ist, endet dieses Vorgehen in v_0.

Eulersche Graphen

Satz

Sei $G = (V, E)$ ein Graph. Paarweise äquivalent sind

(i) *G ist eulersch.*

(ii) *G ist zusammenhängend und* $\forall v \in V : \deg(v)$ *ist gerade.*

(iii) *G ist zusammenhängend und E ist kantendisjunkte Vereinigung von Kreisen.*

Beweis.

$(i) \Rightarrow (ii)$: klar (genauso oft rein wie raus).

$(ii) \Longrightarrow (iii)$: Wähle $v_0 \in V$ und laufe solange noch neue Kanten gefunden werden. Da $\deg(v)$ gerade ist, endet dieses Vorgehen in v_0. Enferne die so gefundenen Kreise und fahre rekursiv fort, bis der Graph leer ist.

Eulersche Graphen

Satz

Sei $G = (V, E)$ ein Graph. Paarweise äquivalent sind

(i) *G ist eulersch.*

(ii) *G ist zusammenhängend und $\forall v \in V : \deg(v)$ ist gerade.*

(iii) *G ist zusammenhängend und E ist kantendisjunkte Vereinigung von Kreisen.*

Beweis.

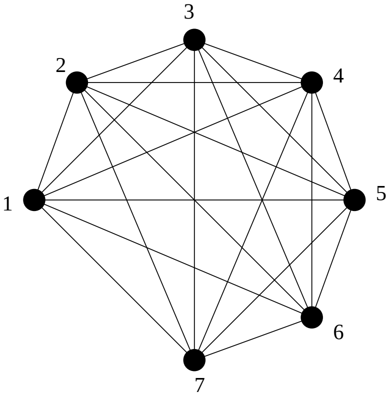
$(i) \Rightarrow (ii)$: klar (genauso oft rein wie raus).

$(ii) \Longrightarrow (iii)$: Wähle $v_0 \in V$ und laufe solange noch neue Kanten gefunden werden. Da $\deg(v)$ gerade ist, endet dieses Vorgehen in v_0. Enferne die so gefundenen Kreise und fahre rekursiv fort, bis der Graph leer ist.

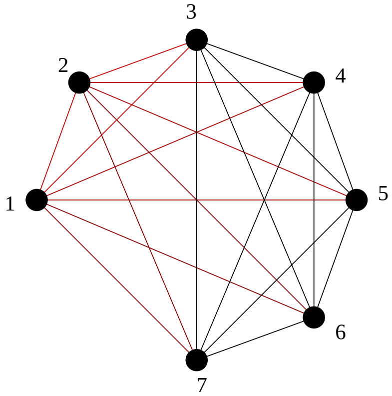
$(iii) \Longrightarrow (i)$: Setze Kreise zu Tour zusammen.

$\square$

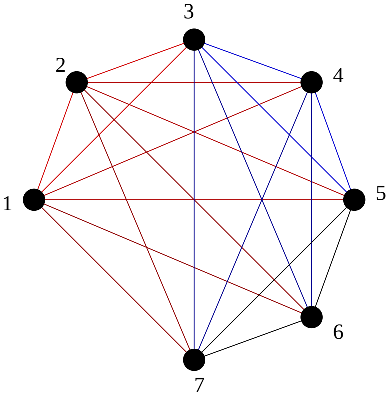
Beispiel:



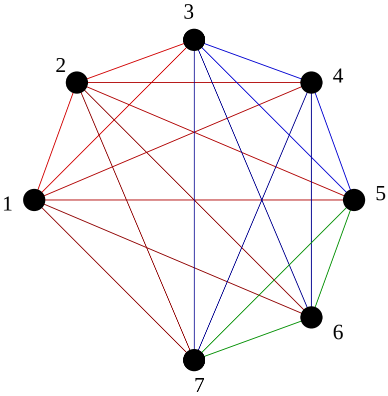
Beispiel:



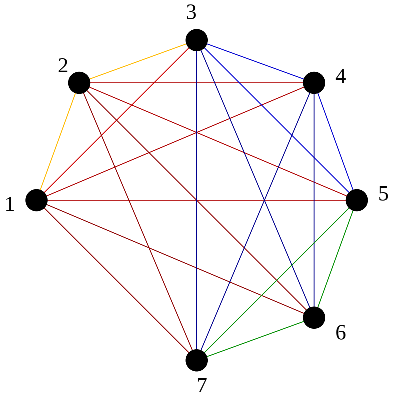
Beispiel:



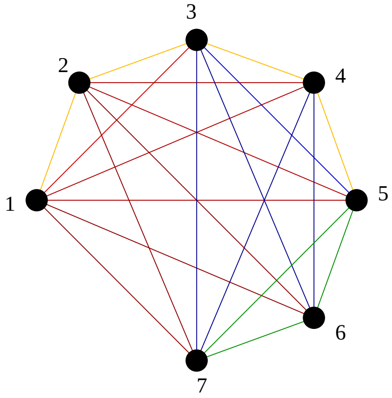
Beispiel:



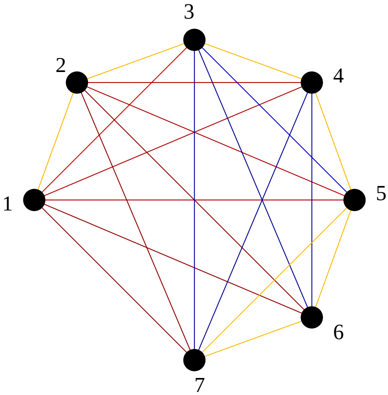
Beispiel:



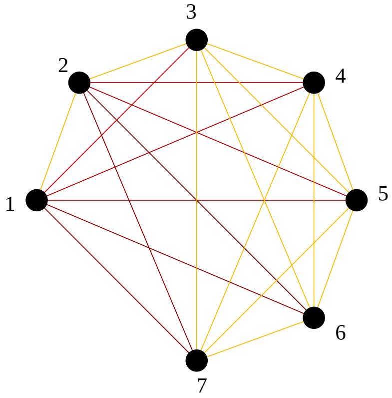
Beispiel:



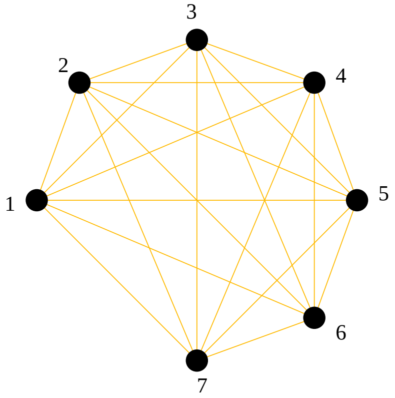
Beispiel:



Beispiel:



Beispiel:



Genauso oft rein wie raus

Satz: Sei $D = (V, A)$ ein gerichteter Graph. Äquivalent sind

(1) D eulersch

(2) D zusammenhängend und $\forall v \in V : \deg^+(v) = \deg^-(v)$

(3) D zusammenhängend und A disjunkte Vereinigung von gerichteten Kreisen

Zweizusammenhang

$G = (V, E)$ heißt k-zusammenhängend ($k \geq 2$), falls

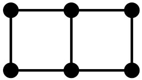
- $|V| \geq k + 1$ und
- beim Löschen von $k - 1$ Knoten bleibt G zusammenhängend

Zweizusammenhang

$G = (V, E)$ heißt k-zusammenhängend ($k \geq 2$), falls

- $|V| \geq k + 1$ und
- beim Löschen von $k - 1$ Knoten bleibt G zusammenhängend

Beispiel



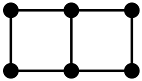
ist 2-zusammenhängend.

Zweizusammenhang

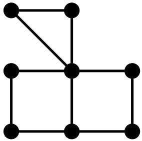
$G = (V, E)$ heißt *k-zusammenhängend* ($k \geq 2$), falls

- $|V| \geq k + 1$ und
- beim Löschen von $k - 1$ Knoten bleibt G zusammenhängend

Beispiel



ist 2-zusammenhängend.



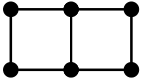
ist nicht 2-zusammenhängend.

Zweizusammenhang

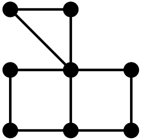
$G = (V, E)$ heißt *k-zusammenhängend* ($k \geq 2$), falls

- $|V| \geq k + 1$ und
- beim Löschen von $k - 1$ Knoten bleibt G zusammenhängend

Beispiel



ist 2-zusammenhängend.



ist nicht 2-zusammenhängend.

Schnittknoten

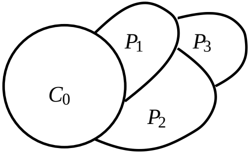
Ohrenzerlegungen

Eine Ohrenzerlegung $(C_0, P_1, P_2, \ldots, P_m)$ besteht aus einem Kreis (C_0) und Pfaden (P_i) beliebiger Länge, die wie folgt angefügt werden:

Ohrenzerlegungen

Eine Ohrenzerlegung $(C_0, P_1, P_2, \ldots, P_m)$ besteht aus einem Kreis (C_0) und Pfaden (P_i) beliebiger Länge, die wie folgt angefügt werden:

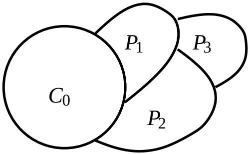
erlaubt:



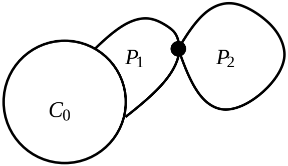
Ohrenzerlegungen

Eine Ohrenzerlegung $(C_0, P_1, P_2, \ldots, P_m)$ besteht aus einem Kreis (C_0) und Pfaden (P_i) beliebiger Länge, die wie folgt angefügt werden:

erlaubt:



nicht erlaubt:



P_2 ist kein Ohr, da P_2 kein Pfad, sondern Kreis ist

Zweizusammenhang und Ohrenzerlegungen

Satz

Sei $G = (V, E)$ Graph. Paarweise äquivalent sind:

(i) *G ist 2-zusammenhängend*

(ii) *je zwei Knoten liegen auf gemeinsamem Kreis*

(iii) *G hat Ohrenzerlegung*

Zweizusammenhang und Ohrenzerlegungen

Satz

Sei $G = (V, E)$ Graph. Paarweise äquivalent sind:

(i) *G ist 2-zusammenhängend*

(ii) *je zwei Knoten liegen auf gemeinsamem Kreis*

(iii) *G hat Ohrenzerlegung*

Beweis.

(ii) $\Rightarrow$ (i): klar.

$\square$

Zweizusammenhang und Ohrenzerlegungen

Satz

Sei $G = (V, E)$ Graph. Paarweise äquivalent sind:

(i) *G ist 2-zusammenhängend*

(ii) *je zwei Knoten liegen auf gemeinsamem Kreis*

(iii) *G hat Ohrenzerlegung*

Beweis.

(ii) $\Rightarrow$ (i): klar.

(i) $\Rightarrow$ (ii): Induktion nach $n :=$dist(u, v).

$\square$

Zweizusammenhang und Ohrenzerlegungen

Satz

Sei $G = (V, E)$ Graph. Paarweise äquivalent sind:

(i) *G ist 2-zusammenhängend*

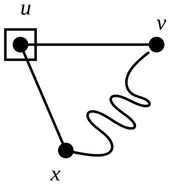
(ii) *je zwei Knoten liegen auf gemeinsamem Kreis*

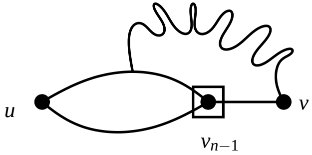
(iii) *G hat Ohrenzerlegung*

Beweis.

(ii) $\Rightarrow$ (i): klar.

(i) $\Rightarrow$ (ii): Induktion nach $n := \mathrm{dist}(u, v)$.





Zweizusammenhang und Ohrenzerlegungen

Satz

Sei $G = (V, E)$ Graph. Paarweise äquivalent sind:

(i) *G ist 2-zusammenhängend*

(ii) *je zwei Knoten liegen auf gemeinsamem Kreis*

(iii) *G hat Ohrenzerlegung*

Beweis.

(ii) $\Rightarrow$ (i): klar.

(i) $\Rightarrow$ (ii): Induktion nach $n :=$dist(u, v).

(i) $\Longleftrightarrow$ (iii): Algorithmisch, siehe Kurstext.

$\square$

Operationen auf Multigraphen I

Einfügen einer Kante

$$G + e' := (V, E \dot{\cup} \{e'\})$$

Operationen auf Multigraphen I

Einfügen einer Kante

$$G + e' := (V, E \dot\cup \{e'\})$$

Entfernen einer Kante

$$G \setminus e := (V, E \setminus \{e\})$$

Operationen auf Multigraphen II

Entfernen eines Knotens

$$G \setminus v := (V \setminus \{v\}, \{e \in E \mid v \notin e\})$$

Operationen auf Multigraphen II

Entfernen eines Knotens

$$G \setminus v := (V \setminus \{v\}, \{e \in E \mid v \notin e\})$$

Unterteilen einer Kante $e = (v, w)$

$$G\%e := (V \dot{\cup} \{u\}, (E \setminus \{e\}) \cup \{(v, u), (u, w)\})$$

Operationen auf Multigraphen III

Kontraktion einer Kante $e = (v, w)$

$$
\begin{aligned}
G/e \quad := \quad & ((V \dot{\cup} \{u\}) \setminus \{v, w\}, \\
& \{e' \in E \mid e' \cap \{v, w\} = \emptyset\} \\
& \cup \{(u, x) \mid (v, x) \in E\} \\
& \cup \{(y, u) \mid (y, w) \in E\})
\end{aligned}
$$

Charakterisierung von Bäumen

Ein kreisfreier und zusammenhängender Graph T heißt Baum.

Satz

Sei $G = (V, E)$ *ein Graph. Paarweise äquivalent sind:*

Charakterisierung von Bäumen

Ein kreisfreier und zusammenhängender Graph T heißt Baum.

Satz

Sei $G = (V, E)$ ein Graph. Paarweise äquivalent sind:

1. *G ist Baum.*

Charakterisierung von Bäumen

Ein kreisfreier und zusammenhängender Graph T heißt Baum.

Satz

Sei $G = (V, E)$ ein Graph. Paarweise äquivalent sind:

1. *G ist Baum.*
2. *G ist kreisfrei und $|E| = |V| - 1$.*

Charakterisierung von Bäumen

Ein kreisfreier und zusammenhängender Graph T heißt Baum.

Satz

Sei $G = (V, E)$ ein Graph. Paarweise äquivalent sind:

1. *G ist Baum.*
2. *G ist kreisfrei und $|E| = |V| - 1$.*
3. *G ist zusammenhängend und $|E| = |V| - 1$.*

Charakterisierung von Bäumen

Ein kreisfreier und zusammenhängender Graph T heißt Baum.

Satz

Sei $G = (V, E)$ ein Graph. Paarweise äquivalent sind:

1. *G ist Baum.*
2. *G ist kreisfrei und $|E| = |V| - 1$.*
3. *G ist zusammenhängend und $|E| = |V| - 1$.*
4. *$\forall v, w \in V : \exists!\ v, w$-Weg in G.*

Charakterisierung von Bäumen

Ein kreisfreier und zusammenhängender Graph T heißt Baum.

Satz

Sei $G = (V, E)$ ein Graph. Paarweise äquivalent sind:

1. *G ist Baum.*
2. *G ist kreisfrei und $|E| = |V| - 1$.*
3. *G ist zusammenhängend und $|E| = |V| - 1$.*
4. $\forall v, w \in V : \exists!\ v, w$*-Weg in G.*
5. *G kreisfrei und $\forall e \notin E : G + e$ hat Kreis.*

Charakterisierung von Bäumen

Ein kreisfreier und zusammenhängender Graph T heißt Baum.

Satz

Sei $G = (V, E)$ ein Graph. Paarweise äquivalent sind:

1. *G ist Baum.*
2. *G ist kreisfrei und $|E| = |V| - 1$.*
3. *G ist zusammenhängend und $|E| = |V| - 1$.*
4. *$\forall v, w \in V : \exists!\ v, w$-Weg in G.*
5. *G kreisfrei und $\forall e \notin E : G + e$ hat Kreis.*
6. *G zusammenhängend und $\forall e \in E : G \setminus e$ unzusammenhängend.*

Existenz von Blättern

Der Beweis der äquivalenten Charakterisierung von Bäumen benutzt an mehreren Stellen das folgende Lemma.

Existenz von Blättern

Der Beweis der äquivalenten Charakterisierung von Bäumen benutzt an mehreren Stellen das folgende Lemma. Ein Knoten $v \in V$ heißt Blatt von V, wenn $\deg(v) = 1$.

Existenz von Blättern

Der Beweis der äquivalenten Charakterisierung von Bäumen benutzt an mehreren Stellen das folgende Lemma. Ein Knoten $v \in V$ heißt Blatt von V, wenn $\deg(v) = 1$.

Lemma

Ein Baum mit mindestens zwei Knoten hat mindestens zwei Blätter.

Existenz von Blättern

Der Beweis der äquivalenten Charakterisierung von Bäumen benutzt an mehreren Stellen das folgende Lemma. Ein Knoten $v \in V$ heißt Blatt von V, wenn $\deg(v) = 1$.

Lemma

Ein Baum mit mindestens zwei Knoten hat mindestens zwei Blätter.

Beweis.

Sei P ein Pfad maximaler Länge in T, seien dessen Endknoten u und v. Dann ist $u \neq v$ und beide müssen Blätter des Graphen sein. □

Isomorphietest von Bäumen

Wenn Bäume gleich gezeichnet sind, sehen wir Ihnen an, ob Sie isomorph sind.

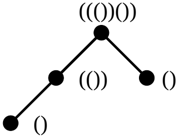
Isomorphietest von Bäumen

Wenn Bäume gleich gezeichnet sind, sehen wir Ihnen an, ob Sie isomorph sind. Wir wollen also eine Vereinbarung treffen, wie ein Baum zu zeichnen ist.

Isomorphietest von Bäumen

Wenn Bäume gleich gezeichnet sind, sehen wir Ihnen an, ob Sie isomorph sind. Wir wollen also eine Vereinbarung treffen, wie ein Baum zu zeichnen ist. Formal ordnen wir dafür jedem Baum einen Code zu.

Isomorphietest von Bäumen



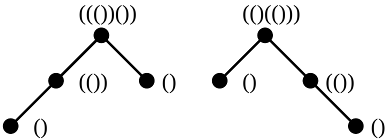
gepfl. Baum

((())())

korrekter Code

((())())

Isomorphietest von Bäumen



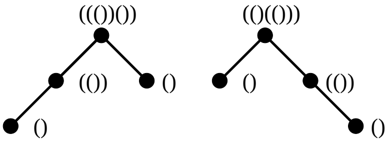
gepfl. Baum

((())()) (()(()))

korrekter Code

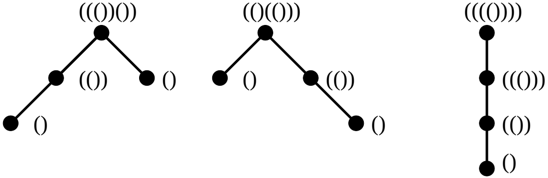
((())())

Isomorphietest von Bäumen



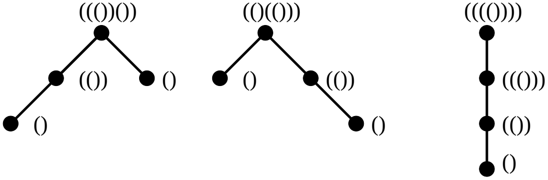
gepfl. Baum		
Wurzelbäumen (sort. Kind. lex.)	((())())	(()(()))
	korrekter Code	vorher nicht korrekt sortiert
	((())())	((())())

Isomorphietest von Bäumen



gepfl. Baum			
Wurzelbäumen (sort. Kind. lex.)	((())())	(()(()))	(((())))
	korrekter Code	vorher nicht korrekt sortiert	
	((())())	((())())	

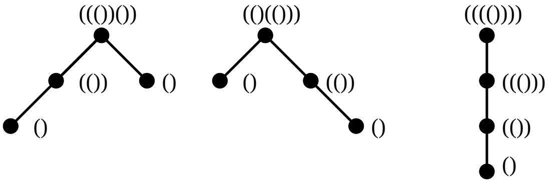
Isomorphietest von Bäumen



gepfl. Baum			
Wurzelbäumen (sort. Kind. lex.)	((())())	(()(()))	(((())))
	korrekter Code	vorher nicht korrekt sortiert	nicht korrekt gewurzelt
Bäumen	((())())	((())())	

Isomorphietest von Bäumen

gepfl. Baum



Wurzelbäumen (sort. Kind. lex.)	((())())	(()(()))	(((())))
	korrekter Code	vorher nicht korrekt sortiert	nicht korrekt gewurzelt
Bäumen (best. Zentrum)	((())())	((())())	((())())

Man zeigt induktiv

Satz

Bäume sind genau dann isomorph, wenn sie den gleichen Code haben.

Hieraus erhalten wir folgendes Verfahren:

Man zeigt induktiv

Satz

Bäume sind genau dann isomorph, wenn sie den gleichen Code haben.

Hieraus erhalten wir folgendes Verfahren:

1. Bestimme Zentrum

Man zeigt induktiv

Satz

Bäume sind genau dann isomorph, wenn sie den gleichen Code haben.

Hieraus erhalten wir folgendes Verfahren:

1. Bestimme Zentrum
2. Erstelle rekursiv Codes für Unterbäume der Zentrumsknoten, lexikographisch geordnet.

Man zeigt induktiv

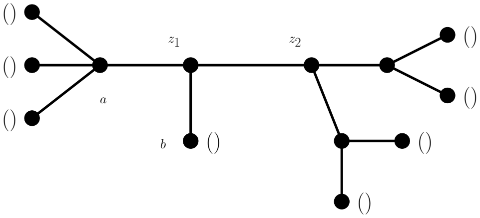
Satz

Bäume sind genau dann isomorph, wenn sie den gleichen Code haben.

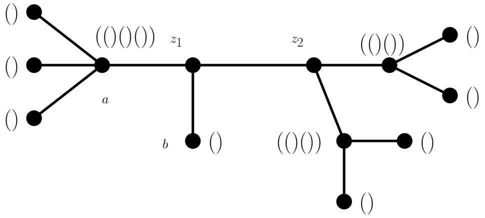
Hieraus erhalten wir folgendes Verfahren:

1. Bestimme Zentrum
2. Erstelle rekursiv Codes für Unterbäume der Zentrumsknoten, lexikographisch geordnet.
3. Zentrumsknoten mit kleinerem Code ist Wurzel

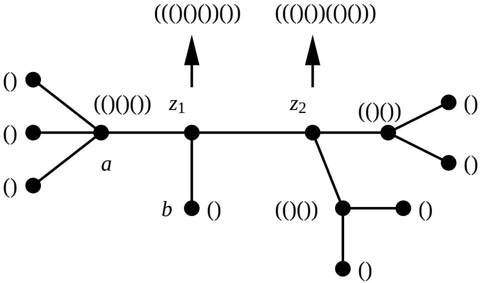
Beispiel



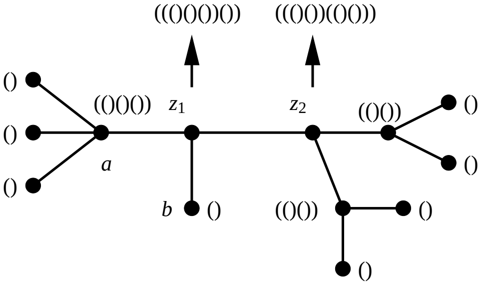
Beispiel



Beispiel



Beispiel



Es gilt: $c(z_1) \preceq c(z_2)$, also ist z_1 Wurzel und der Gesamtcode lautet

$$(\underbrace{((()())(()()))}_{z_2}\underbrace{(()()())}_{a}\underbrace{()}_{b})$$