Karlsruher Institut für Technologie

Algorithmen II
Übung am 22.01.2013

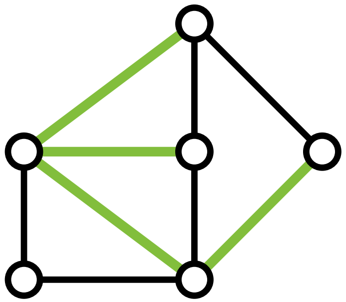
Approximation und Parametrisierung

INSTITUT FÜR THEORETISCHE INFORMATIK · PROF. DR. DOROTHEA WAGNER

Minimale Schnittbasis

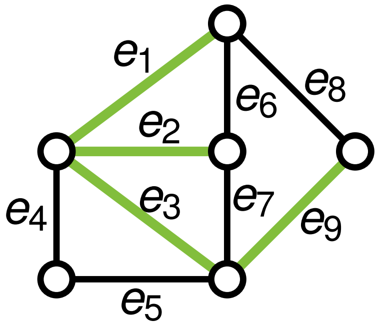
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Der Schnittraum



Kantenmenge

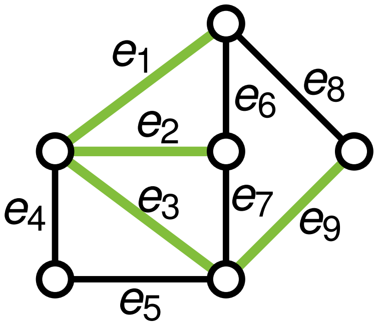
Der Schnittraum



Kantenmenge

in Vektordarstellung $\begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$

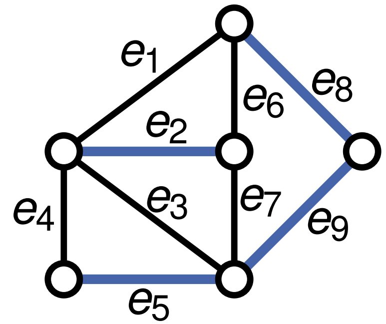
Der Schnittraum



Kantenmenge

in Vektordarstellung

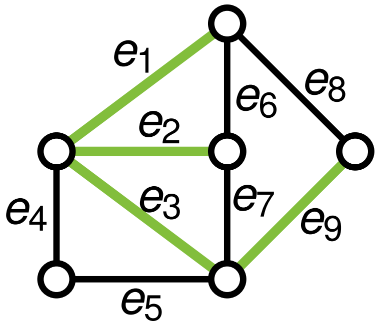
$$\begin{pmatrix}1\\1\\1\\0\\0\\0\\0\\0\\1\end{pmatrix} + \begin{pmatrix}0\\1\\0\\0\\1\\0\\0\\1\\1\end{pmatrix}$$



symmetrische Differenz

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

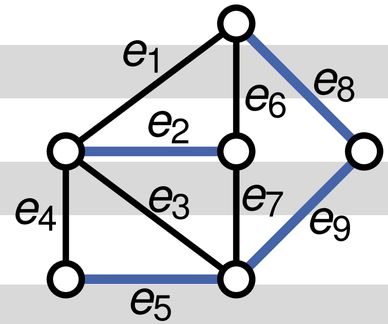
Der Schnittraum



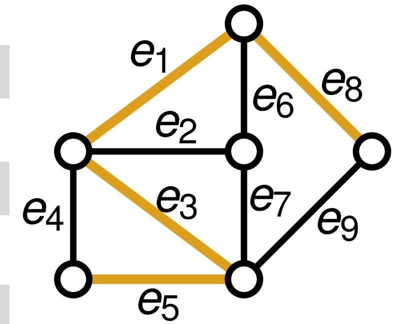
Kantenmenge

in Vektordarstellung

$$\begin{pmatrix}1\\1\\1\\0\\0\\0\\0\\0\\1\end{pmatrix} + \begin{pmatrix}0\\1\\0\\0\\1\\0\\0\\1\\1\end{pmatrix}$$



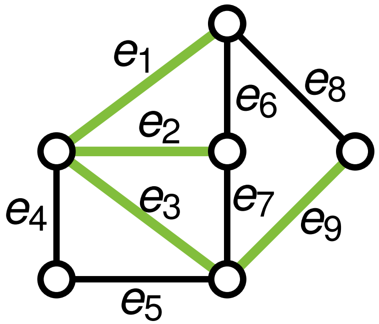
$$= \begin{pmatrix}1\\0\\1\\0\\1\\0\\0\\1\\0\end{pmatrix}$$



symmetrische Differenz

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

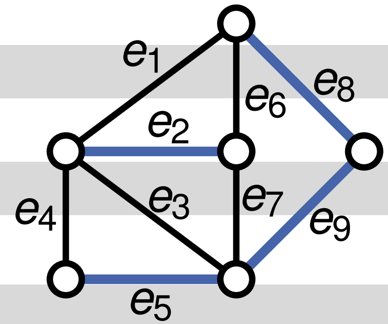
Der Schnittraum



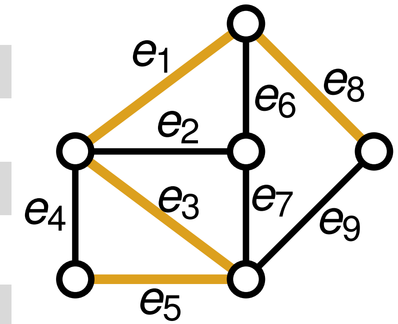
Kantenmenge

in Vektordarstellung

$$\begin{pmatrix}1\\1\\1\\0\\0\\0\\0\\0\\1\end{pmatrix} + \begin{pmatrix}0\\1\\0\\0\\1\\0\\0\\1\\1\end{pmatrix}$$



$$= \begin{pmatrix}1\\0\\1\\0\\1\\0\\0\\1\\0\end{pmatrix}$$

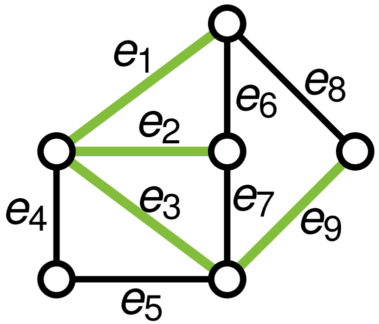


symmetrische Differenz

Menge aller Kantenmengen bildet Vektorraum. Interessante Unterräume:

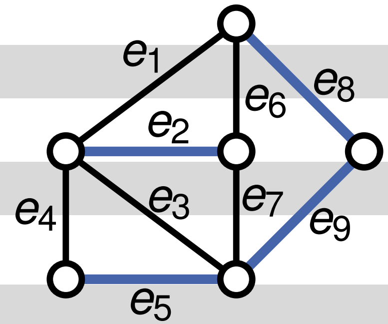
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

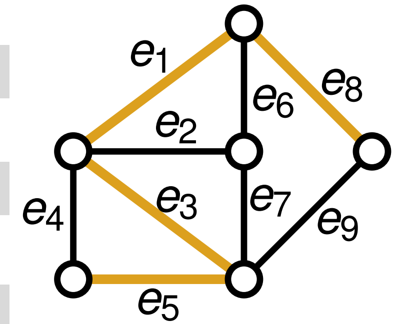
Der Schnittraum



Kantenmenge
in Vektordarstellung

$$\begin{pmatrix}1\\1\\1\\0\\0\\0\\0\\0\\1\end{pmatrix} + \begin{pmatrix}0\\1\\0\\0\\1\\0\\0\\1\\1\end{pmatrix} = \begin{pmatrix}1\\0\\1\\0\\1\\0\\0\\1\\0\end{pmatrix}$$



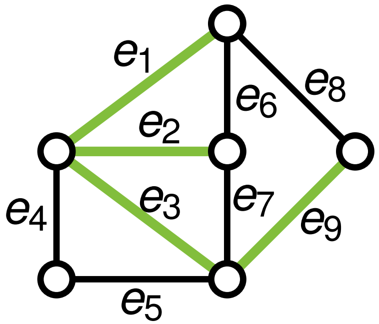


symmetrische Differenz

Menge aller Kantenmengen bildet Vektorraum. Interessante Unterräume:

- Aus der Vorlesung: Kreisraum
- heute: Schnittraum

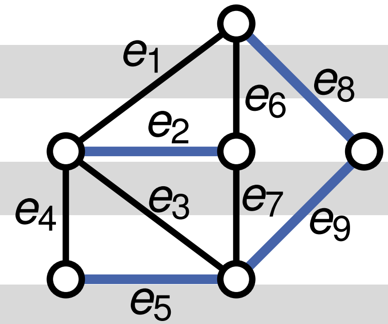
Der Schnittraum

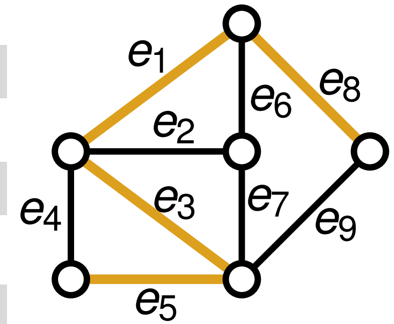


Kantenmenge

in Vektordarstellung

$$\begin{pmatrix}1\\1\\1\\0\\0\\0\\0\\0\\1\end{pmatrix} + \begin{pmatrix}0\\1\\0\\0\\1\\0\\0\\1\\1\end{pmatrix} = \begin{pmatrix}1\\0\\1\\0\\1\\0\\0\\1\\0\end{pmatrix}$$

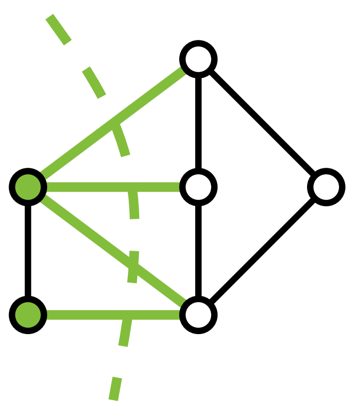




symmetrische Differenz

Menge aller Kantenmengen bildet Vektorraum. Interessante Unterräume:

- Aus der Vorlesung: Kreisraum
- heute: Schnittraum

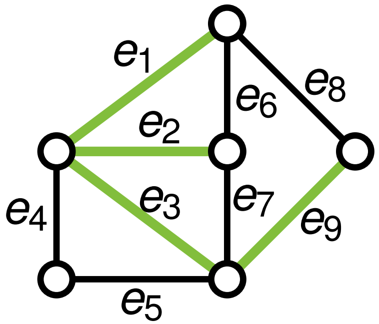


Mögliche Darstellungen eines Schnitts:

- Partition der Knoten in zwei Teilmengen S und $V \setminus S$.
- Menge der Kanten von S nach $V \setminus S$.

Achtung: Nicht jede Kantenmenge bildet Schnitt!

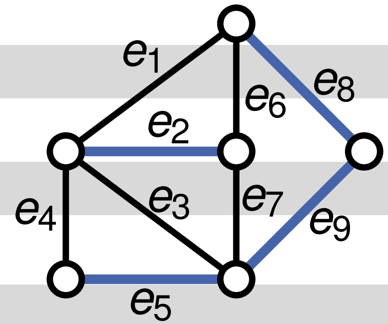
Der Schnittraum



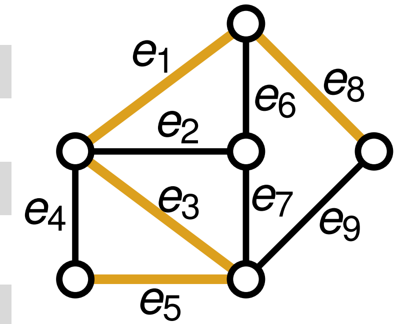
Kantenmenge

in Vektordarstellung

$$\begin{pmatrix}1\\1\\1\\0\\0\\0\\0\\0\\1\end{pmatrix} + \begin{pmatrix}0\\1\\0\\0\\1\\0\\0\\1\\1\end{pmatrix}$$



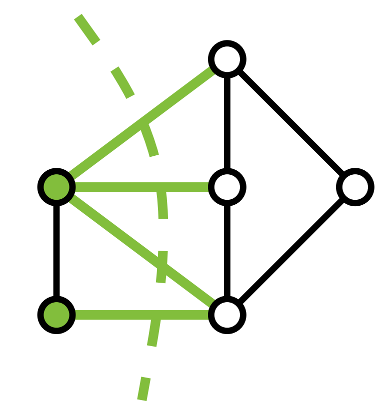
$$= \begin{pmatrix}1\\0\\1\\0\\1\\0\\0\\1\\0\end{pmatrix}$$



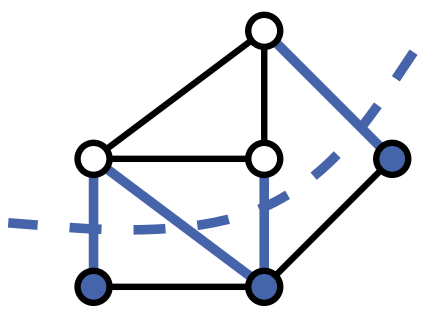
symmetrische Differenz

Menge aller Kantenmengen bildet Vektorraum. Interessante Unterräume:

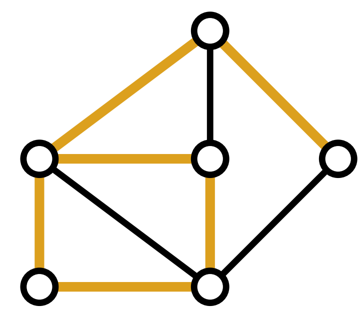
- Aus der Vorlesung: Kreisraum
- heute: Schnittraum



+

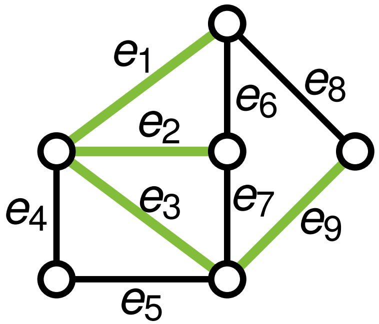


=



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

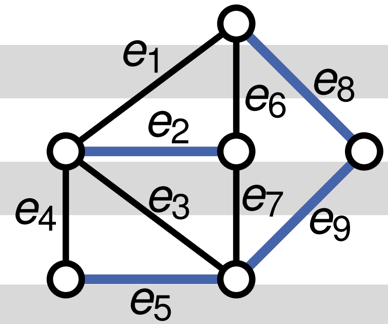
Der Schnittraum

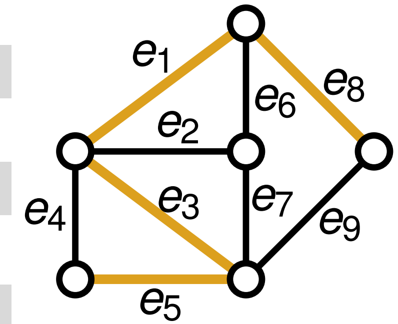


Kantenmenge

in Vektordarstellung

$$\begin{pmatrix}1\\1\\1\\0\\0\\0\\0\\0\\1\end{pmatrix} + \begin{pmatrix}0\\1\\0\\0\\1\\0\\0\\1\\1\end{pmatrix} = \begin{pmatrix}1\\0\\1\\0\\1\\0\\0\\1\\0\end{pmatrix}$$

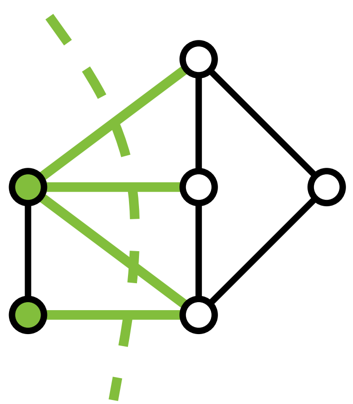




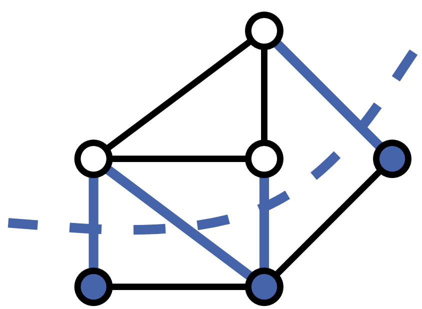
symmetrische Differenz

Menge aller Kantenmengen bildet Vektorraum. Interessante Unterräume:

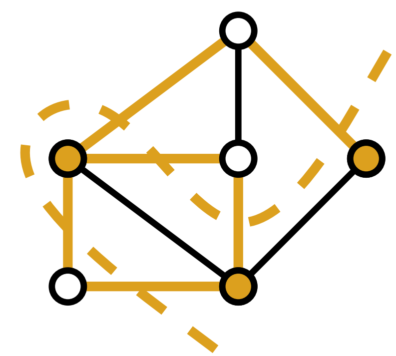
- Aus der Vorlesung: Kreisraum
- heute: Schnittraum



+



=

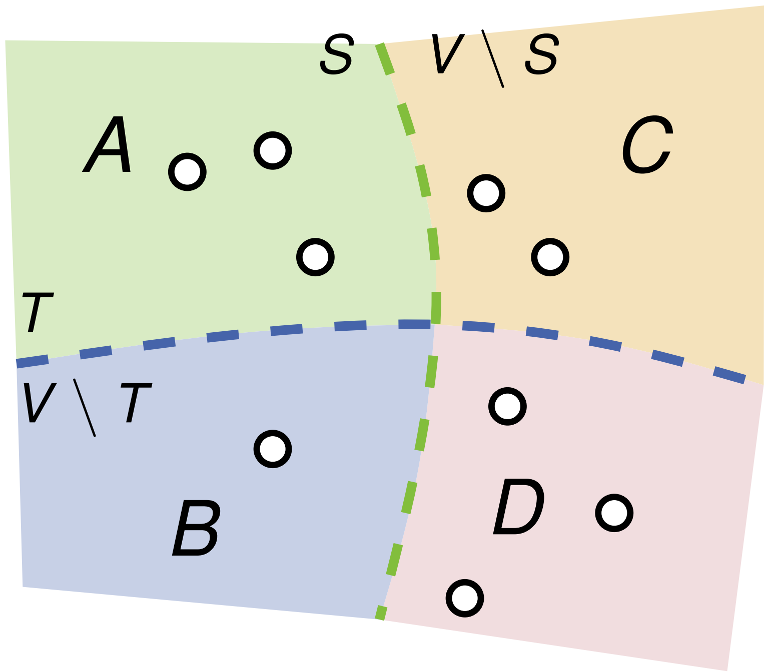


Problem 1 (a) Formulieren Sie die symmetrische Differenz in Partitionsdarstellung.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 1 (a)

(a) Formulieren Sie die Partitionsdarstellung des Schnitts $s_3 = s_1 + s_2$ (mit $s_1 = (S, V \setminus S)$, $s_2 = (T, V \setminus T)$) in Abhängigkeit von S und T.



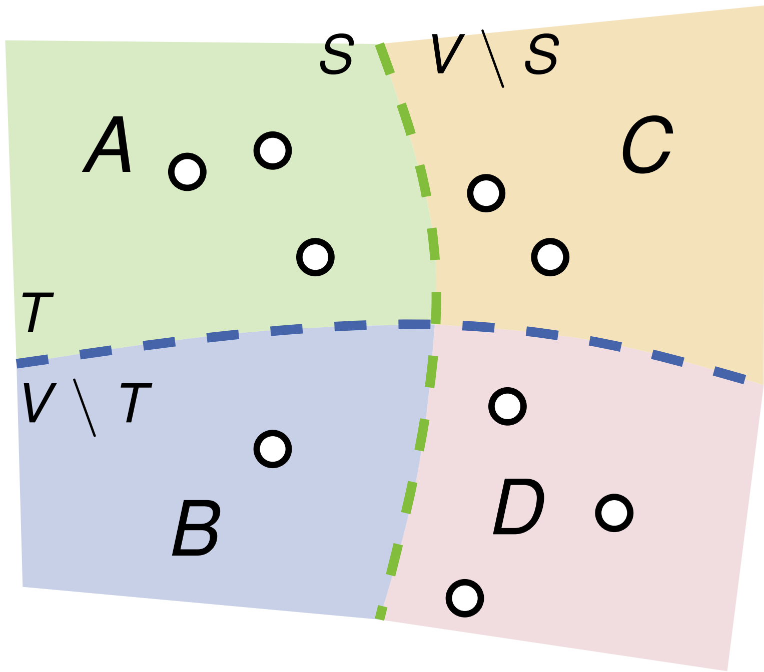
Interessante Knotenmengen:

- $A = S \cap T$
- $B = S \cap (V \setminus T)$
- $C = (V \setminus S) \cap T$
- $D = (V \setminus S) \cap (V \setminus T)$

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 1 (a)

(a) Formulieren Sie die Partitionsdarstellung des Schnitts $s_3 = s_1 + s_2$ (mit $s_1 = (S, V \setminus S)$, $s_2 = (T, V \setminus T)$) in Abhängigkeit von S und T.



Interessante Knotenmengen:

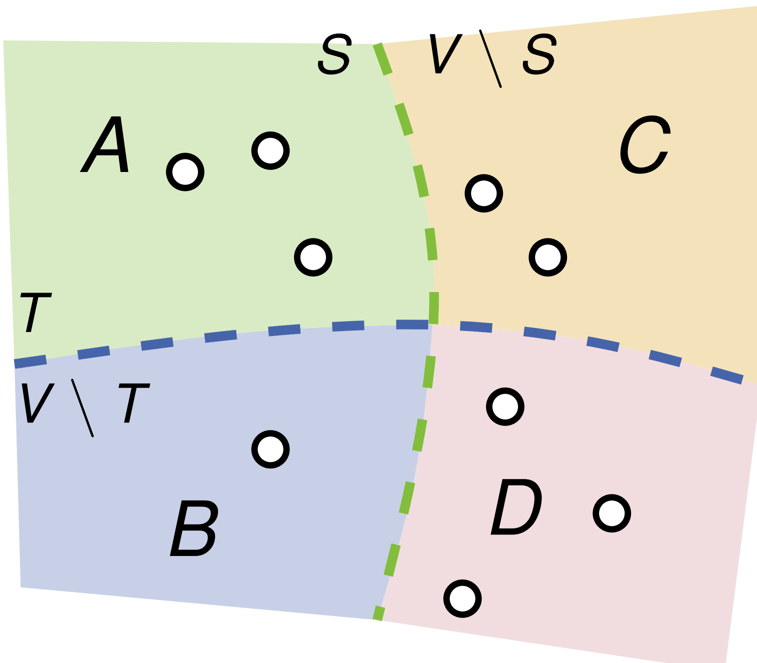
- $A = S \cap T$
- $B = S \cap (V \setminus T)$
- $C = (V \setminus S) \cap T$
- $D = (V \setminus S) \cap (V \setminus T)$

Welche Kanten sind in s_3 enthalten?

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 1 (a)

(a) Formulieren Sie die Partitionsdarstellung des Schnitts $s_3 = s_1 + s_2$ (mit $s_1 = (S, V \setminus S)$, $s_2 = (T, V \setminus T)$) in Abhängigkeit von S und T.



Interessante Knotenmengen:

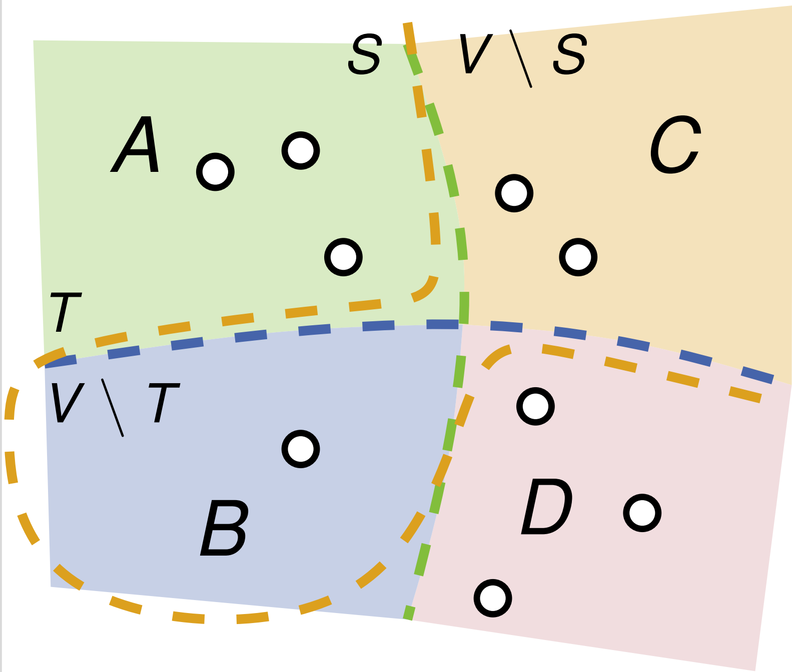
- $A = S \cap T$
- $B = S \cap (V \setminus T)$
- $C = (V \setminus S) \cap T$
- $D = (V \setminus S) \cap (V \setminus T)$

Welche Kanten sind in s_3 enthalten?

- Alle, die genau einen der beiden Schnitte kreuzen.

Problem 1 (a)

(a) Formulieren Sie die Partitionsdarstellung des Schnitts $s_3 = s_1 + s_2$ (mit $s_1 = (S, V \setminus S)$, $s_2 = (T, V \setminus T)$) in Abhängigkeit von S und T.



Interessante Knotenmengen:

- $A = S \cap T$
- $B = S \cap (V \setminus T)$
- $C = (V \setminus S) \cap T$
- $D = (V \setminus S) \cap (V \setminus T)$

Welche Kanten sind in s_3 enthalten?

- Alle, die genau einen der beiden Schnitte kreuzen.
- $\Rightarrow s_3 = (A \cup D, B \cup C)$

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 1 (b)

(b) Zeigen Sie: Für je zwei Knoten $u, v \in V$ und jede Basis B des Schnittraums gilt, dass mindestens ein Schnitt aus B die Knoten u und v trennt.

Beobachtung:

Im Schnittraum gibt es einen Schnitt, der u und v trennt: $s_3 = \left(\{u\}, V \setminus \{u\}\right)$

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 1 (b)

(b) Zeigen Sie: Für je zwei Knoten $u, v \in V$ und jede Basis B des Schnittraums gilt, dass mindestens ein Schnitt aus B die Knoten u und v trennt.

Beobachtung:

Im Schnittraum gibt es einen Schnitt, der u und v trennt: $s_3 = \left(\{u\}, V \setminus \{u\}\right)$

Behauptung:

Falls u und v von $s_3 = s_1 + s_2$ getrennt werden, dann auch schon von s_1 oder s_2.

Problem 1 (b)

(b) Zeigen Sie: Für je zwei Knoten $u, v \in V$ und jede Basis B des Schnittraums gilt, dass mindestens ein Schnitt aus B die Knoten u und v trennt.

Beobachtung:

Im Schnittraum gibt es einen Schnitt, der u und v trennt: $s_3 = \big(\{u\}, V \setminus \{u\}\big)$

Behauptung:

Falls u und v von $s_3 = s_1 + s_2$ getrennt werden, dann auch schon von s_1 oder s_2.

Folgerung:

Da s_3 als Linearkombination von Schnitten aus B dargestellt werden kann, muss mindestens einer dieser Schnitte u und v trennen.

Problem 1 (b)

(b) Zeigen Sie: Für je zwei Knoten $u, v \in V$ und jede Basis B des Schnittraums gilt, dass mindestens ein Schnitt aus B die Knoten u und v trennt.

Beobachtung:

Im Schnittraum gibt es einen Schnitt, der u und v trennt: $s_3 = \left(\{u\}, V \setminus \{u\}\right)$

Behauptung:

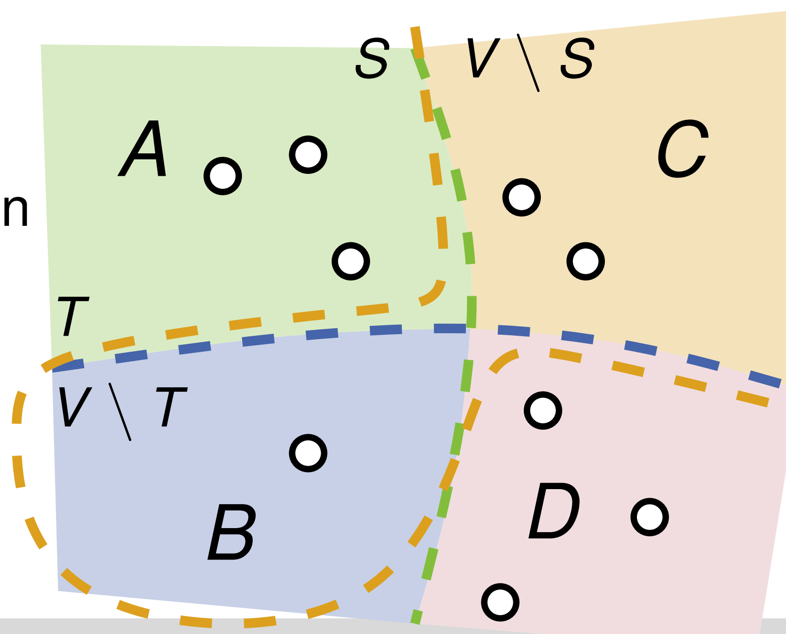
Falls u und v von $s_3 = s_1 + s_2$ getrennt werden, dann auch schon von s_1 oder s_2.

Folgerung:

Da s_3 als Linearkombination von Schnitten aus B dargestellt werden kann, muss mindestens einer dieser Schnitte u und v trennen.

Beweis der Behauptung:

- Betrachte alle Möglichkeiten, wie u und v von $s_3 = s_1 + s_2$ getrennt werden können.



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 1 (b)

(b) Zeigen Sie: Für je zwei Knoten $u, v \in V$ und jede Basis B des Schnittraums gilt, dass mindestens ein Schnitt aus B die Knoten u und v trennt.

Beobachtung:

Im Schnittraum gibt es einen Schnitt, der u und v trennt: $s_3 = \left(\{u\}, V \setminus \{u\}\right)$

Behauptung:

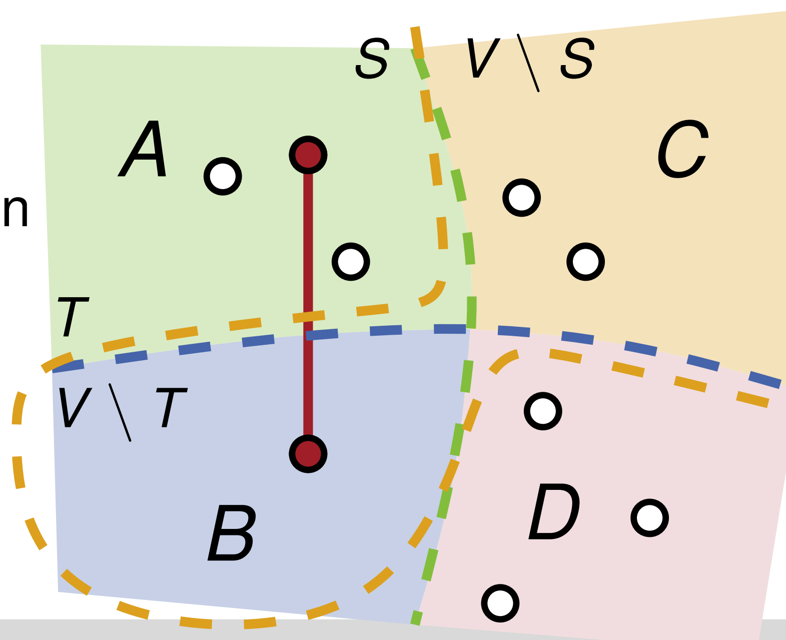
Falls u und v von $s_3 = s_1 + s_2$ getrennt werden, dann auch schon von s_1 oder s_2.

Folgerung:

Da s_3 als Linearkombination von Schnitten aus B dargestellt werden kann, muss mindestens einer dieser Schnitte u und v trennen.

Beweis der Behauptung:

- Betrachte alle Möglichkeiten, wie u und v von $s_3 = s_1 + s_2$ getrennt werden können.



Problem 1 (b)

(b) Zeigen Sie: Für je zwei Knoten $u, v \in V$ und jede Basis B des Schnittraums gilt, dass mindestens ein Schnitt aus B die Knoten u und v trennt.

Beobachtung:

Im Schnittraum gibt es einen Schnitt, der u und v trennt: $s_3 = \big(\{u\}, V \setminus \{u\}\big)$

Behauptung:

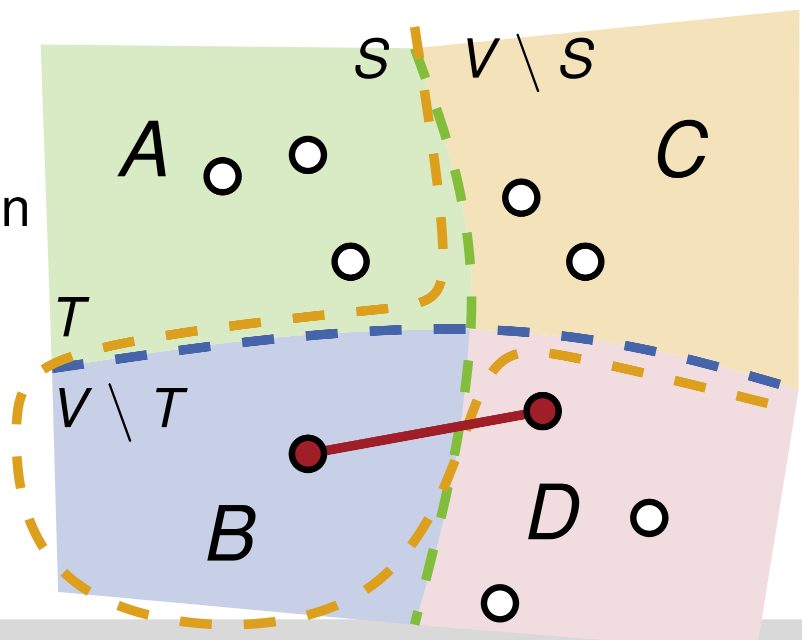
Falls u und v von $s_3 = s_1 + s_2$ getrennt werden, dann auch schon von s_1 oder s_2.

Folgerung:

Da s_3 als Linearkombination von Schnitten aus B dargestellt werden kann, muss mindestens einer dieser Schnitte u und v trennen.

Beweis der Behauptung:

- Betrachte alle Möglichkeiten, wie u und v von $s_3 = s_1 + s_2$ getrennt werden können.



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 1 (b)

(b) Zeigen Sie: Für je zwei Knoten $u, v \in V$ und jede Basis B des Schnittraums gilt, dass mindestens ein Schnitt aus B die Knoten u und v trennt.

Beobachtung:

Im Schnittraum gibt es einen Schnitt, der u und v trennt: $s_3 = \big(\{u\}, V \setminus \{u\}\big)$

Behauptung:

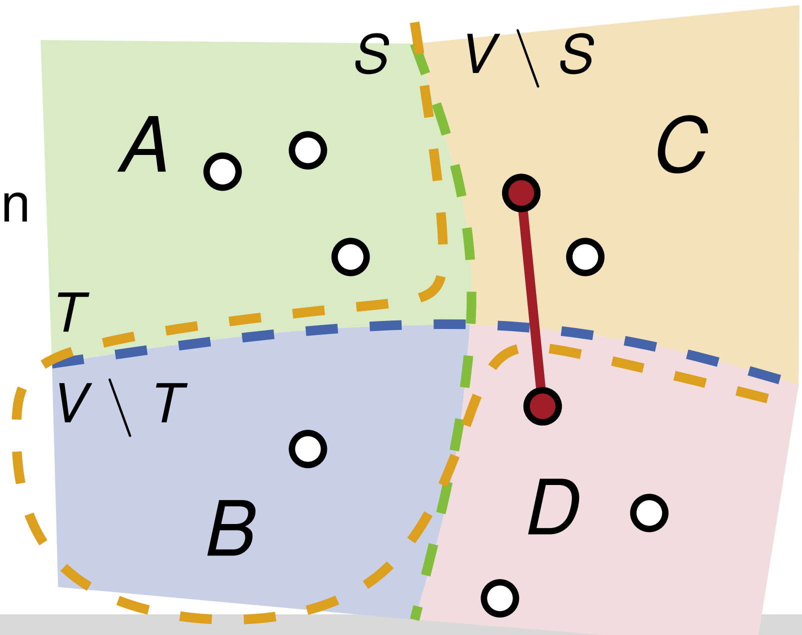
Falls u und v von $s_3 = s_1 + s_2$ getrennt werden, dann auch schon von s_1 oder s_2.

Folgerung:

Da s_3 als Linearkombination von Schnitten aus B dargestellt werden kann, muss mindestens einer dieser Schnitte u und v trennen.

Beweis der Behauptung:

- Betrachte alle Möglichkeiten, wie u und v von $s_3 = s_1 + s_2$ getrennt werden können.



Problem 1 (b)

(b) Zeigen Sie: Für je zwei Knoten $u, v \in V$ und jede Basis B des Schnittraums gilt, dass mindestens ein Schnitt aus B die Knoten u und v trennt.

Beobachtung:

Im Schnittraum gibt es einen Schnitt, der u und v trennt: $s_3 = \left(\{u\}, V \setminus \{u\}\right)$

Behauptung:

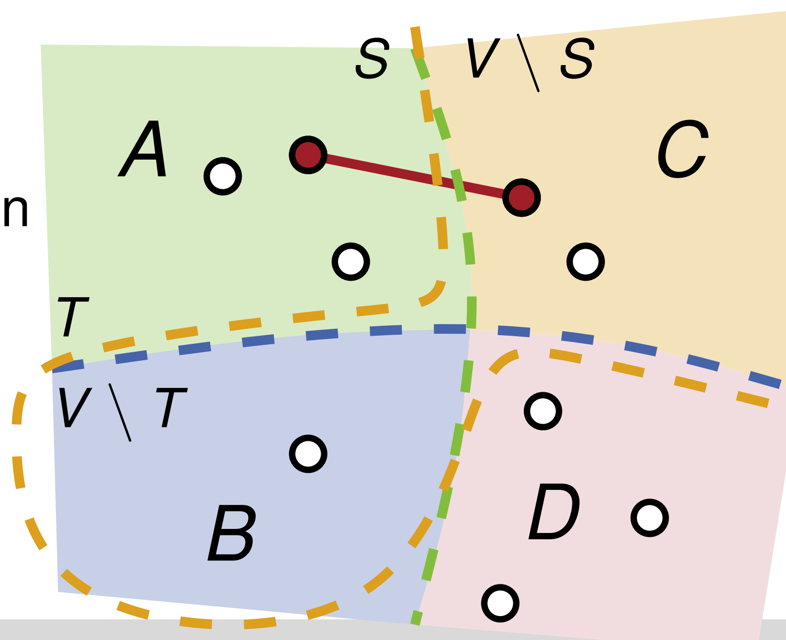
Falls u und v von $s_3 = s_1 + s_2$ getrennt werden, dann auch schon von s_1 oder s_2.

Folgerung:

Da s_3 als Linearkombination von Schnitten aus B dargestellt werden kann, muss mindestens einer dieser Schnitte u und v trennen.

Beweis der Behauptung:

- Betrachte alle Möglichkeiten, wie u und v von $s_3 = s_1 + s_2$ getrennt werden können.



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 1 (c)

Sei $B = \{b_1, \ldots, b_d\}$ eine Basis des Schnittraums von G.

- Gewicht eines Schnittes b_i: $c(b_i)$ = #Kanten, die b_i kreuzen

Problem 1 (c)

Sei $B = \{b_1, \ldots, b_d\}$ eine Basis des Schnittraums von G.

- Gewicht eines Schnittes b_i: $c(b_i)$ = #Kanten, die b_i kreuzen
- Gewicht der Basis B: $c(B) = \sum_{i=1}^{d} c(b_i)$

Problem 1 (c)

Sei $B = \{b_1, \ldots, b_d\}$ eine Basis des Schnittraums von G.

- Gewicht eines Schnittes b_i: $c(b_i)$ = #Kanten, die b_i kreuzen
- Gewicht der Basis B: $c(B) = \sum_{i=1}^{d} c(b_i)$

Problem: MIN-SCHNITT-BASIS
Finde eine Basis des Schnittraums von G mit minimalem Gewicht.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 1 (c)

Sei $B = \{b_1, \ldots, b_d\}$ eine Basis des Schnittraums von G.

- Gewicht eines Schnittes b_i: $c(b_i)$ = #Kanten, die b_i kreuzen
- Gewicht der Basis B: $c(B) = \sum_{i=1}^{d} c(b_i)$

Problem: MIN-SCHNITT-BASIS
Finde eine Basis des Schnittraums von G mit minimalem Gewicht.

APPROX-MIN-SCHNITT-BASIS(G)
  Wähle einen Knoten $v \in V$
  $B' \leftarrow \emptyset$
  forall $v' \in V \setminus \{v\}$ **do**
    $b_{v'} \leftarrow \{e \in E | e \text{ inzident zu } v'\}$
    $B' \leftarrow B' \cup \{b_{v'}\}$
  Return B'

(c) Zeigen Sie: APPROX-MIN-SCHNITT-BASIS ist ein Approximationsalgorithmus mit relativer Gütegarantie 2.

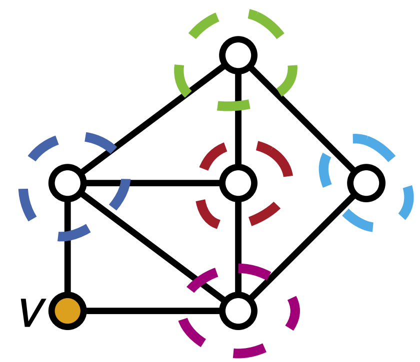
Problem 1 (c)

Sei $B = \{b_1, \ldots, b_d\}$ eine Basis des Schnittraums von G.

- Gewicht eines Schnittes b_i: $c(b_i)$ = #Kanten, die b_i kreuzen
- Gewicht der Basis B: $c(B) = \sum_{i=1}^{d} c(b_i)$

Problem: MIN-SCHNITT-BASIS
Finde eine Basis des Schnittraums von G mit minimalem Gewicht.

APPROX-MIN-SCHNITT-BASIS(G)
  Wähle einen Knoten $v \in V$
  $B' \leftarrow \emptyset$
  forall $v' \in V \setminus \{v\}$ **do**
    $b_{v'} \leftarrow \{e \in E | e \text{ inzident zu } v'\}$
    $B' \leftarrow B' \cup \{b_{v'}\}$
  Return B'



(c) Zeigen Sie: APPROX-MIN-SCHNITT-BASIS ist ein Approximationsalgorithmus mit relativer Gütegarantie 2.

Problem 1 (c)

APPROX-MIN-SCHNITT-BASIS(G)

  Wähle einen Knoten $v \in V$

  $B' \leftarrow \emptyset$

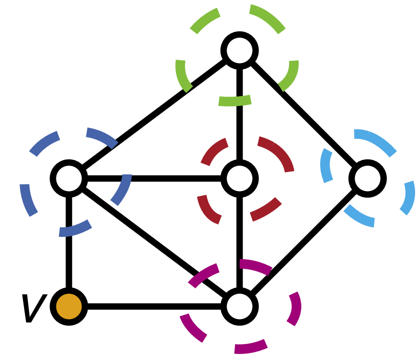
  forall $v' \in V \setminus \{v\}$ **do**

    $b_{v'} \leftarrow \{e \in E | e \text{ inzident zu } v'\}$

    $B' \leftarrow B' \cup \{b_{v'}\}$

  Return B'

Wie groß ist $c(B')$?



(c) Zeigen Sie: APPROX-MIN-SCHNITT-BASIS ist ein Approximationsalgorithmus mit relativer Gütegarantie 2.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 1 (c)

APPROX-MIN-SCHNITT-BASIS(G)

Wähle einen Knoten $v \in V$

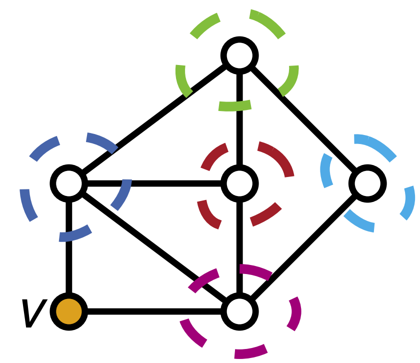
$B' \leftarrow \emptyset$

forall $v' \in V \setminus \{v\}$ **do**

$b_{v'} \leftarrow \{e \in E | e \text{ inzident zu } v'\}$

$B' \leftarrow B' \cup \{b_{v'}\}$

Return B'



(c) Zeigen Sie: APPROX-MIN-SCHNITT-BASIS ist ein Approximationsalgorithmus mit relativer Gütegarantie 2.

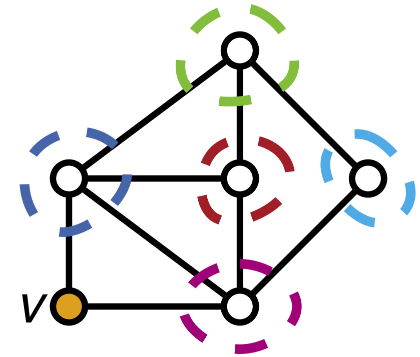
Wie groß ist $c(B')$?

- Kanten inzident zu v sind in einem Schnitt enthalten.
- Alle anderen Kanten in zwei Schnitten.

$\Rightarrow c(B') = 2m - \deg(v) \leq 2m$ $(m = |E|)$

Problem 1 (c)

APPROX-MIN-SCHNITT-BASIS(G)
Wähle einen Knoten $v \in V$
$B' \leftarrow \emptyset$
forall $v' \in V \setminus \{v\}$ **do**
$b_{v'} \leftarrow \{e \in E | e \text{ inzident zu } v'\}$
$B' \leftarrow B' \cup \{b_{v'}\}$
Return B'



(c) Zeigen Sie: APPROX-MIN-SCHNITT-BASIS ist ein Approximationsalgorithmus mit relativer Gütegarantie 2.

Wie groß ist $c(B')$?

- Kanten inzident zu v sind in einem Schnitt enthalten.
- Alle anderen Kanten in zwei Schnitten.

$\Rightarrow c(B') = 2m - \deg(v) \leq 2m$ $\qquad (m = |E|)$

Wie groß ist $c(B)$ für eine minimale Basis B?

Problem 1 (c)

APPROX-MIN-SCHNITT-BASIS(G)

Wähle einen Knoten $v \in V$

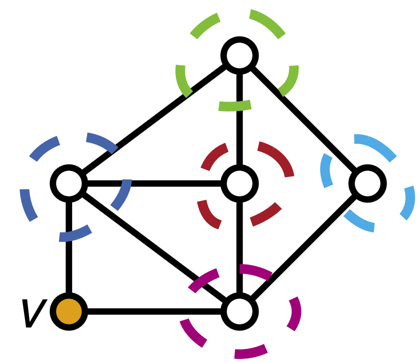
$B' \leftarrow \emptyset$

forall $v' \in V \setminus \{v\}$ **do**

$b_{v'} \leftarrow \{e \in E | e \text{ inzident zu } v'\}$

$B' \leftarrow B' \cup \{b_{v'}\}$

Return B'



(c) Zeigen Sie: APPROX-MIN-SCHNITT-BASIS ist ein Approximationsalgorithmus mit relativer Gütegarantie 2.

Wie groß ist $c(B')$?

- Kanten inzident zu v sind in einem Schnitt enthalten.
- Alle anderen Kanten in zwei Schnitten.

$\Rightarrow c(B') = 2m - \deg(v) \leq 2m$ $(m = |E|)$

Wie groß ist $c(B)$ für eine minimale Basis B?

- Betrachte beliebige Kante $\{u, w\}$.
- Wegen (b) gilt: in B gibt es einen Schnitt, der u und w trennt.
- Jede Kante ist in mindestens einem Schnitt in B enthalten.

$\Rightarrow c(B) \geq m \Rightarrow$ APPROX-MIN-SCHNITT-BASIS hat Gütegarantie 2.

Organisatorisches

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Organisatorisches

Evaluation der Übung

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Organisatorisches

Evaluation der Übung

Klausuren

Hauptklausur:

- Freitag 01.03.2013 um 11:00 Uhr
- Bearbeitungszeit: 2 Stunden
- Anmeldung über das Studienportal, Anmeldezeitraum startet in wenigen Tagen und geht bis 22.02.2013.
- Hörsaalbelegung wird auf der Homepage bekannt gegeben (natürlich erst nach dem Ende des Anmeldezeitraums).

Nachklausur:

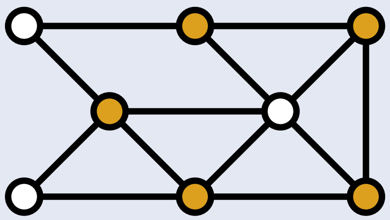
- Mittwoch 02.10.2013 um 14:00 Uhr
- Weitere Informationen gibt es zu gegebener Zeit auf der Homepage.

Vertex Cover

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 2

Problem: VERTEX COVER
Finde *Vertex Cover* $V' \subseteq V$ mit $|V'| \leq k$. V' heißt Vertex Cover genau dann, wenn für jede Kante $\{v, w\} \in E$ gilt $v \in V'$ oder $w \in V'$.
(ist $\mathcal{NP}$-Schwer)

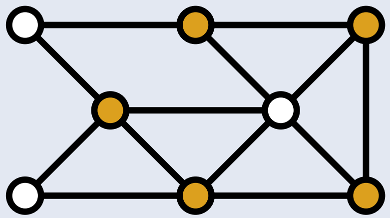


Problem 2: Geben Sie einen Algorithmus für VERTEX COVER an, der eine Approximationsgüte von 2 hat.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 2

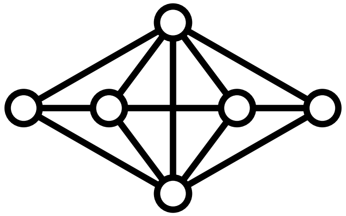
Problem: VERTEX COVER
Finde *Vertex Cover* $V' \subseteq V$ mit $|V'| \leq k$. V' heißt Vertex Cover genau dann, wenn für jede Kante $\{v, w\} \in E$ gilt $v \in V'$ oder $w \in V'$.
(ist $\mathcal{NP}$-Schwer)



Problem 2: Geben Sie einen Algorithmus für VERTEX COVER an, der eine Approximationsgüte von 2 hat.

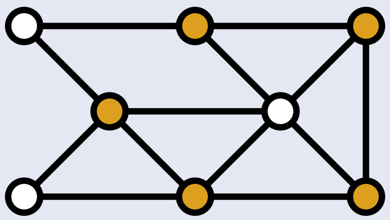
Idee: Führe iterativ folgende Schritte aus:

- Wähle beliebige Kante $\{u, v\}$
- Füge u **und** v zum Vertex Cover hinzu und lösche sie aus dem Graph.



Problem 2

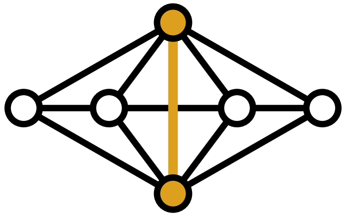
Problem: VERTEX COVER
Finde *Vertex Cover* $V' \subseteq V$ mit $|V'| \leq k$. V' heißt Vertex Cover genau dann, wenn für jede Kante $\{v, w\} \in E$ gilt $v \in V'$ oder $w \in V'$. (ist $\mathcal{NP}$-Schwer)



Problem 2: Geben Sie einen Algorithmus für VERTEX COVER an, der eine Approximationsgüte von 2 hat.

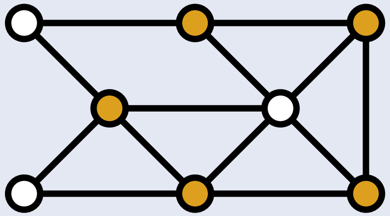
Idee: Führe iterativ folgende Schritte aus:

- Wähle beliebige Kante $\{u, v\}$
- Füge u **und** v zum Vertex Cover hinzu und lösche sie aus dem Graph.



Problem 2

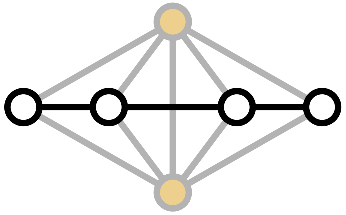
Problem: VERTEX COVER
Finde *Vertex Cover* $V' \subseteq V$ mit $|V'| \leq k$. V' heißt Vertex Cover genau dann, wenn für jede Kante $\{v, w\} \in E$ gilt $v \in V'$ oder $w \in V'$.
(ist $\mathcal{NP}$-Schwer)



Problem 2: Geben Sie einen Algorithmus für VERTEX COVER an, der eine Approximationsgüte von 2 hat.

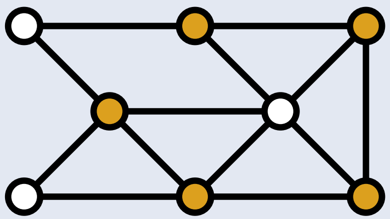
Idee: Führe iterativ folgende Schritte aus:

- Wähle beliebige Kante $\{u, v\}$
- Füge u **und** v zum Vertex Cover hinzu und lösche sie aus dem Graph.



Problem 2

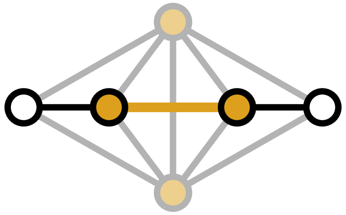
Problem: VERTEX COVER
Finde *Vertex Cover* $V' \subseteq V$ mit $|V'| \leq k$. V' heißt Vertex Cover genau dann, wenn für jede Kante $\{v, w\} \in E$ gilt $v \in V'$ oder $w \in V'$. (ist $\mathcal{NP}$-Schwer)



Problem 2: Geben Sie einen Algorithmus für VERTEX COVER an, der eine Approximationsgüte von 2 hat.

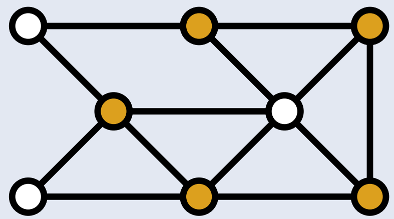
Idee: Führe iterativ folgende Schritte aus:

- Wähle beliebige Kante $\{u, v\}$
- Füge u **und** v zum Vertex Cover hinzu und lösche sie aus dem Graph.



Problem 2

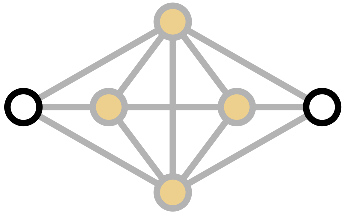
Problem: VERTEX COVER
Finde *Vertex Cover* $V' \subseteq V$ mit $|V'| \leq k$. V' heißt Vertex Cover genau dann, wenn für jede Kante $\{v, w\} \in E$ gilt $v \in V'$ oder $w \in V'$. (ist $\mathcal{NP}$-Schwer)



Problem 2: Geben Sie einen Algorithmus für VERTEX COVER an, der eine Approximationsgüte von 2 hat.

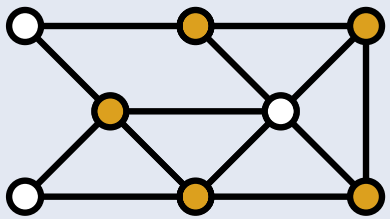
Idee: Führe iterativ folgende Schritte aus:

- Wähle beliebige Kante $\{u, v\}$
- Füge u **und** v zum Vertex Cover hinzu und lösche sie aus dem Graph.



Problem 2

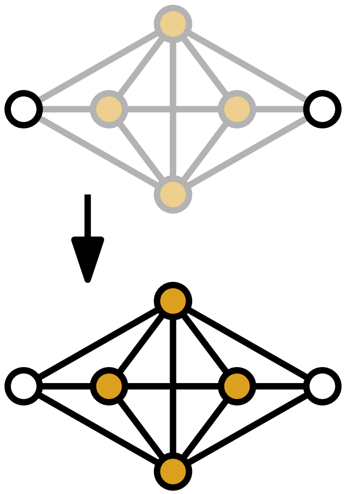
Problem: VERTEX COVER
Finde *Vertex Cover* $V' \subseteq V$ mit $|V'| \leq k$. V' heißt Vertex Cover genau dann, wenn für jede Kante $\{v, w\} \in E$ gilt $v \in V'$ oder $w \in V'$.
(ist $\mathcal{NP}$-Schwer)



Problem 2: Geben Sie einen Algorithmus für VERTEX COVER an, der eine Approximationsgüte von 2 hat.

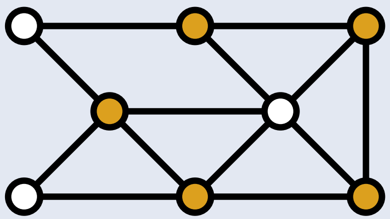
Idee: Führe iterativ folgende Schritte aus:

- Wähle beliebige Kante $\{u, v\}$
- Füge u **und** v zum Vertex Cover hinzu und lösche sie aus dem Graph.



Problem 2

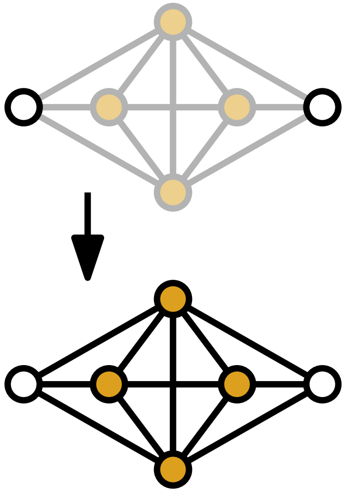
Problem: VERTEX COVER
Finde *Vertex Cover* $V' \subseteq V$ mit $|V'| \leq k$. V' heißt Vertex Cover genau dann, wenn für jede Kante $\{v, w\} \in E$ gilt $v \in V'$ oder $w \in V'$.
(ist $\mathcal{NP}$-Schwer)



Problem 2: Geben Sie einen Algorithmus für VERTEX COVER an, der eine Approximationsgüte von 2 hat.

Idee: Führe iterativ folgende Schritte aus:

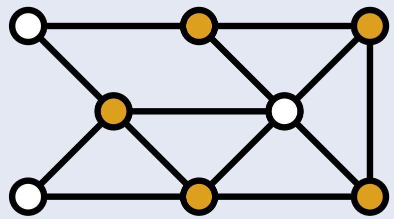
- Wähle beliebige Kante $\{u, v\}$
- Füge u **und** v zum Vertex Cover hinzu und lösche sie aus dem Graph.



Approximationsgüte

Problem 2

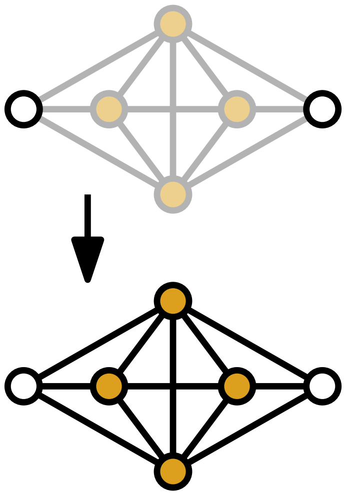
Problem: VERTEX COVER
Finde *Vertex Cover* $V' \subseteq V$ mit $|V'| \leq k$. V' heißt Vertex Cover genau dann, wenn für jede Kante $\{v, w\} \in E$ gilt $v \in V'$ oder $w \in V'$.
(ist $\mathcal{NP}$-Schwer)



Problem 2: Geben Sie einen Algorithmus für VERTEX COVER an, der eine Approximationsgüte von 2 hat.

Idee: Führe iterativ folgende Schritte aus:

- Wähle beliebige Kante $\{u, v\}$
- Füge u **und** v zum Vertex Cover hinzu und lösche sie aus dem Graph.



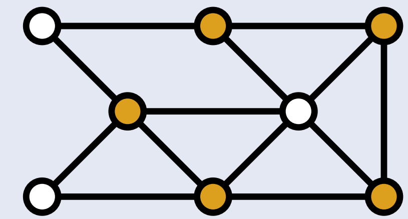
Approximationsgüte

- Seien $e_1, \ldots, e_k$ die vom Algorithmus ausgewählten Kanten.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 2

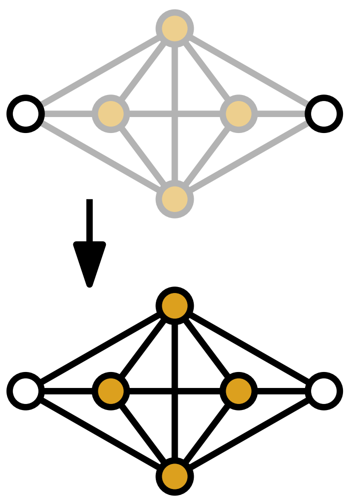
Problem: VERTEX COVER
Finde *Vertex Cover* $V' \subseteq V$ mit $|V'| \leq k$. V' heißt Vertex Cover genau dann, wenn für jede Kante $\{v, w\} \in E$ gilt $v \in V'$ oder $w \in V'$.
(ist $\mathcal{NP}$-Schwer)



Problem 2: Geben Sie einen Algorithmus für VERTEX COVER an, der eine Approximationsgüte von 2 hat.

Idee: Führe iterativ folgende Schritte aus:

- Wähle beliebige Kante $\{u, v\}$
- Füge u **und** v zum Vertex Cover hinzu und lösche sie aus dem Graph.



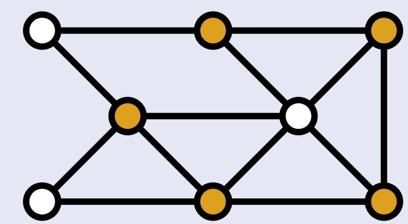
Approximationsgüte

- Seien $e_1, \ldots, e_k$ die vom Algorithmus ausgewählten Kanten.
- Der Algorithmus liefert ein Vertex Cover der Größe $2k$.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problem 2

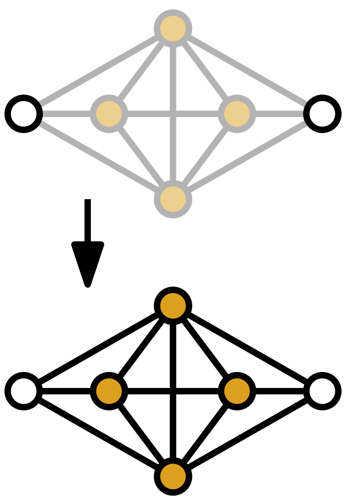
Problem: VERTEX COVER
Finde *Vertex Cover* $V' \subseteq V$ mit $|V'| \leq k$. V' heißt Vertex Cover genau dann, wenn für jede Kante $\{v, w\} \in E$ gilt $v \in V'$ oder $w \in V'$.
(ist $\mathcal{NP}$-Schwer)



Problem 2: Geben Sie einen Algorithmus für VERTEX COVER an, der eine Approximationsgüte von 2 hat.

Idee: Führe iterativ folgende Schritte aus:

- Wähle beliebige Kante $\{u, v\}$
- Füge u **und** v zum Vertex Cover hinzu und lösche sie aus dem Graph.

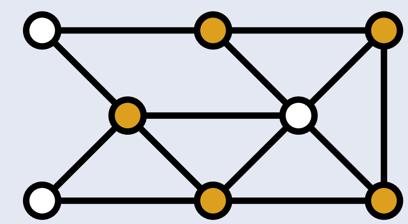


Approximationsgüte

- Seien $e_1, \ldots, e_k$ die vom Algorithmus ausgewählten Kanten.
- Der Algorithmus liefert ein Vertex Cover der Größe $2k$.
- Jedes Vertex Cover enthält mindestens k Knoten, da es keinen Knoten gibt, der mehr als eine der Kanten $e_1, \ldots, e_k$ abdeckt.

Problem 2

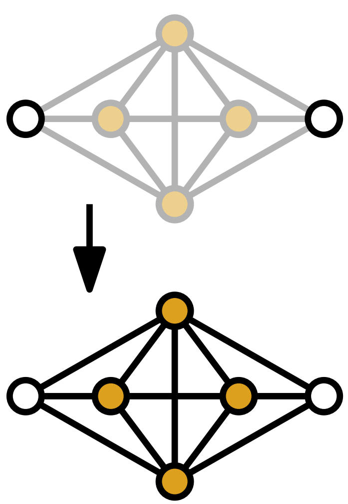
Problem: VERTEX COVER
Finde *Vertex Cover* $V' \subseteq V$ mit $|V'| \leq k$. V' heißt Vertex Cover genau dann, wenn für jede Kante $\{v, w\} \in E$ gilt $v \in V'$ oder $w \in V'$. (ist $\mathcal{NP}$-Schwer)



Problem 2: Geben Sie einen Algorithmus für VERTEX COVER an, der eine Approximationsgüte von 2 hat.

Idee: Führe iterativ folgende Schritte aus:

- Wähle beliebige Kante $\{u, v\}$
- Füge u **und** v zum Vertex Cover hinzu und lösche sie aus dem Graph.



Approximationsgüte

- Seien $e_1, \ldots, e_k$ die vom Algorithmus ausgewählten Kanten.
- Der Algorithmus liefert ein Vertex Cover der Größe $2k$.
- Jedes Vertex Cover enthält mindestens k Knoten, da es keinen Knoten gibt, der mehr als eine der Kanten $e_1, \ldots, e_k$ abdeckt.

$\Rightarrow$ Approximationsgüte 2

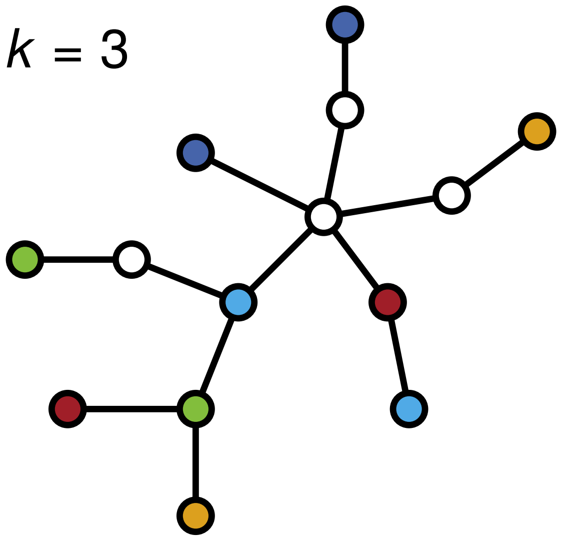
FPT-Algorithmus für MULTICUT

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

MULTICUT in Bäumen

Eingabe:
Baum $T = (V, E)$, eine Menge von Knotenpaaren $H \subseteq \binom{V}{2}$, sowie Parameter k.

$k = 3$

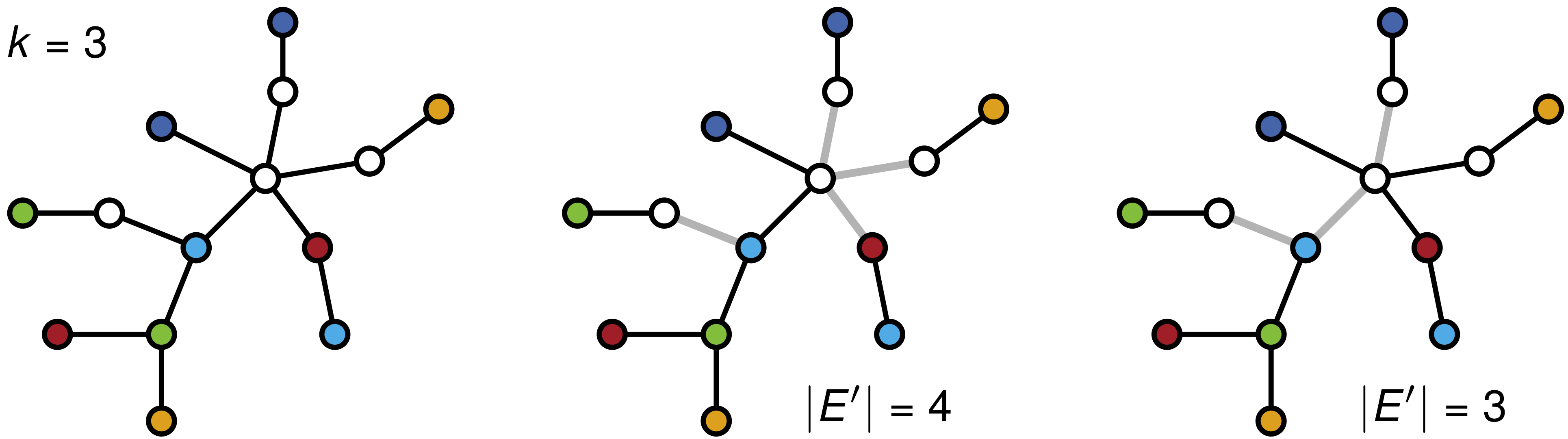


Gibt es eine Kantenmenge $E' \subseteq E$ mit $|E'| \leq k$, die alle Paare trennt?

MULTICUT in Bäumen

Eingabe:

Baum $T = (V, E)$, eine Menge von Knotenpaaren $H \subseteq \binom{V}{2}$, sowie Parameter k.

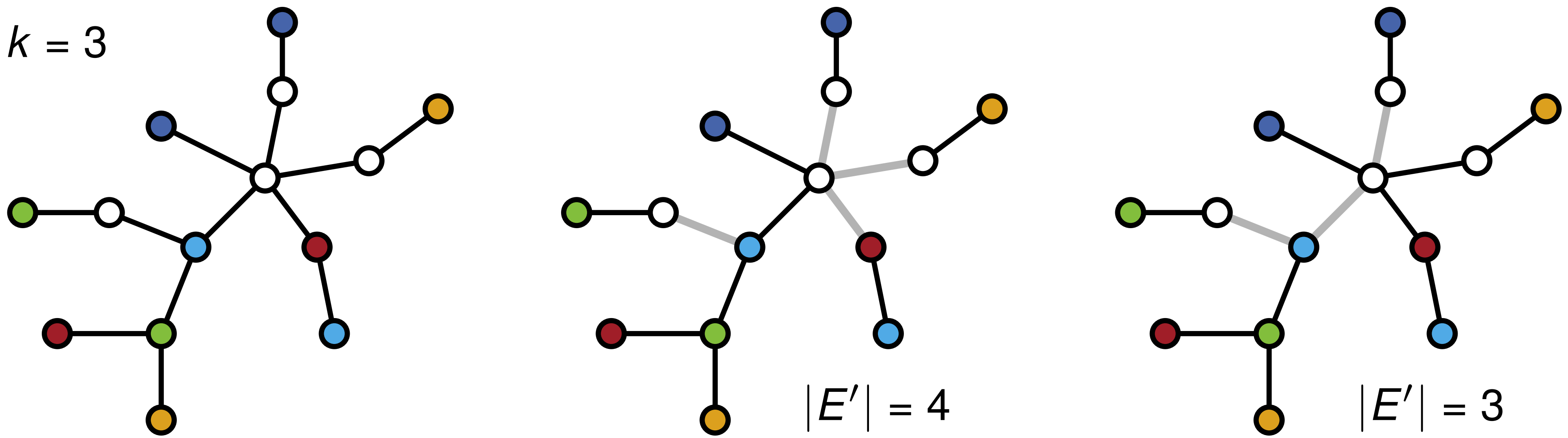


Gibt es eine Kantenmenge $E' \subseteq E$ mit $|E'| \leq k$, die alle Paare trennt?

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

MULTICUT in Bäumen

Eingabe:
Baum $T = (V, E)$, eine Menge von Knotenpaaren $H \subseteq \binom{V}{2}$, sowie Parameter k.



Gibt es eine Kantenmenge $E' \subseteq E$ mit $|E'| \leq k$, die alle Paare trennt?

Problem 3:
Zeigen Sie, dass MULTICUT auf Bäumen Fixed Parameter Tractable bezüglich k ist.

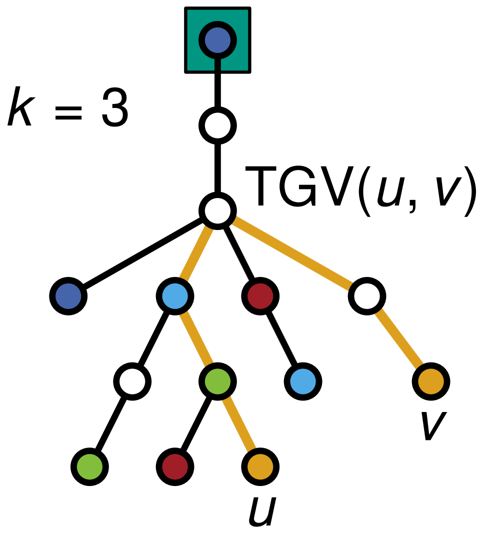
Erinnerung:
Ein Problem ist FPT, wenn es in $O(\mathcal{C}(k) \cdot p(n))$ Zeit gelöst werden kann.
($\mathcal{C}(k)$ ist berechenbar und hängt nur von k ab, $p(n)$ ist ein Polynom in der Eingabegröße n)

Entscheidungsbaum

Ziel: Finde ein Kantenpaar, sodass mindestens eine der beiden Kanten im Schnitt enthalten sein muss. $\rightarrow$ Binärer Entscheidungsbaum

Entscheidungsbaum

Ziel: Finde ein Kantenpaar, sodass mindestens eine der beiden Kanten im Schnitt enthalten sein muss. $\rightarrow$ Binärer Entscheidungsbaum



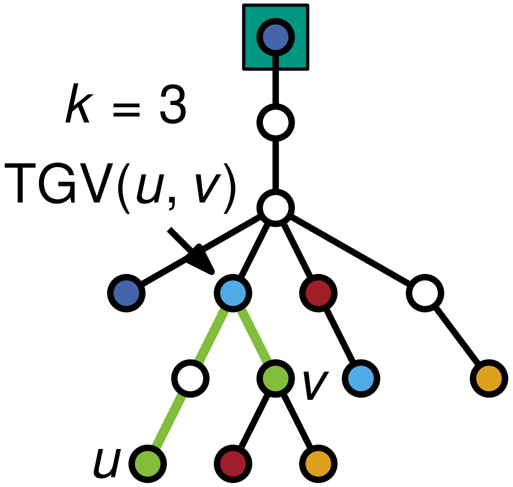
Tiefster gemeinsamer Vorgänger

- Wähle beliebigen Knoten als Wurzel.
- Der Tiefste gemeinsame Vorgänger TGV(u, v) zweier Knoten u und v ist der höchste Knoten auf dem Pfad $\pi(u, v)$ zwischen u und v.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Entscheidungsbaum

Ziel: Finde ein Kantenpaar, sodass mindestens eine der beiden Kanten im Schnitt enthalten sein muss. $\rightarrow$ Binärer Entscheidungsbaum



Tiefster gemeinsamer Vorgänger

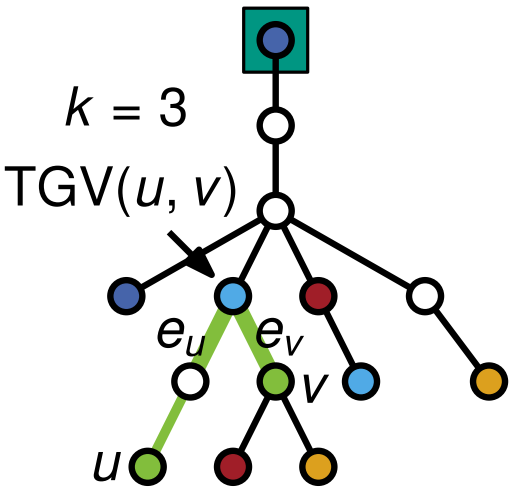
- Wähle beliebigen Knoten als Wurzel.
- Der Tiefste gemeinsame Vorgänger TGV(u, v) zweier Knoten u und v ist der höchste Knoten auf dem Pfad $\pi(u, v)$ zwischen u und v.

Betrachte Knotenpaar $\{u, v\} \in H$, sodass TGV(u, v) möglichst tief ist.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Entscheidungsbaum

Ziel: Finde ein Kantenpaar, sodass mindestens eine der beiden Kanten im Schnitt enthalten sein muss. $\rightarrow$ Binärer Entscheidungsbaum



Tiefster gemeinsamer Vorgänger

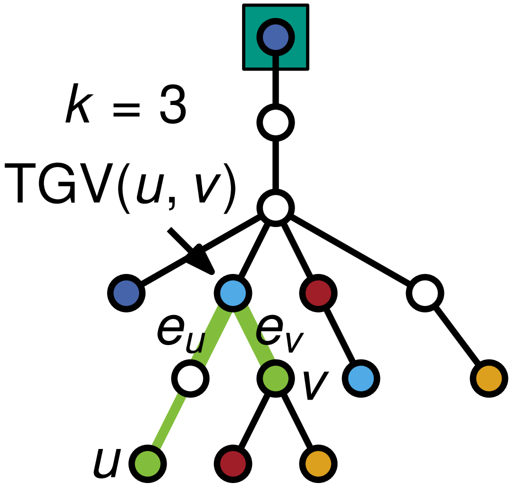
- Wähle beliebigen Knoten als Wurzel.
- Der Tiefste gemeinsame Vorgänger TGV(u, v) zweier Knoten u und v ist der höchste Knoten auf dem Pfad $\pi(u, v)$ zwischen u und v.

Betrachte Knotenpaar $\{u, v\} \in H$, sodass TGV(u, v) möglichst tief ist.

Behauptung: Eine der beiden Kanten e_u und e_v inzident zu TGV(u, v) auf $\pi(u, v)$ ist in jedem minimalen Schnitt enthalten.

Entscheidungsbaum

Ziel: Finde ein Kantenpaar, sodass mindestens eine der beiden Kanten im Schnitt enthalten sein muss. $\rightarrow$ Binärer Entscheidungsbaum



Tiefster gemeinsamer Vorgänger

- Wähle beliebigen Knoten als Wurzel.
- Der Tiefste gemeinsame Vorgänger TGV(u, v) zweier Knoten u und v ist der höchste Knoten auf dem Pfad $\pi(u, v)$ zwischen u und v.

Betrachte Knotenpaar $\{u, v\} \in H$, sodass TGV(u, v) möglichst tief ist.

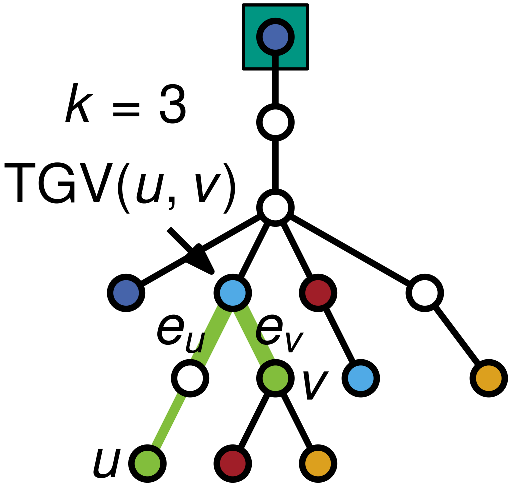
Behauptung: Eine der beiden Kanten e_u und e_v inzident zu TGV(u, v) auf $\pi(u, v)$ ist in jedem minimalen Schnitt enthalten.

Beweis:

- Eine Kante auf $\pi(u, v)$ muss im Schnitt enthalten sein um u von v zu trennen.

Entscheidungsbaum

Ziel: Finde ein Kantenpaar, sodass mindestens eine der beiden Kanten im Schnitt enthalten sein muss. $\rightarrow$ Binärer Entscheidungsbaum



Tiefster gemeinsamer Vorgänger

- Wähle beliebigen Knoten als Wurzel.
- Der Tiefste gemeinsame Vorgänger TGV(u, v) zweier Knoten u und v ist der höchste Knoten auf dem Pfad $\pi(u, v)$ zwischen u und v.

Betrachte Knotenpaar $\{u, v\} \in H$, sodass TGV(u, v) möglichst tief ist.

Behauptung: Eine der beiden Kanten e_u und e_v inzident zu TGV(u, v) auf $\pi(u, v)$ ist in jedem minimalen Schnitt enthalten.

Beweis:

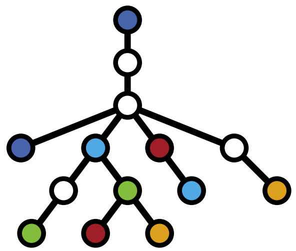
- Eine Kante auf $\pi(u, v)$ muss im Schnitt enthalten sein um u von v zu trennen.
- Keine Kante auf $\pi(u, \mathrm{TGV}(u, v))$ trennt ein Knotenpaar in H, dass nicht von e_u getrennt wird. (analog für e_v)

$\Rightarrow$ Entweder e_u oder e_v müssen gewählt werden.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Beispiel

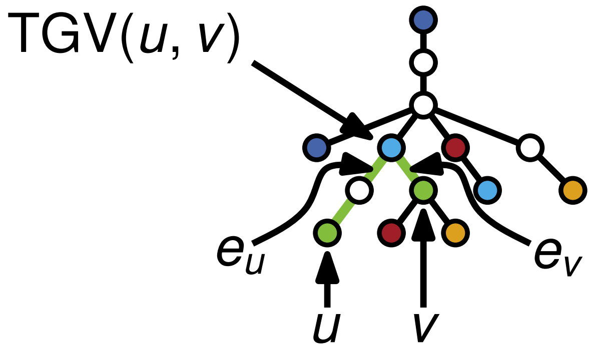
$k = 3$



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

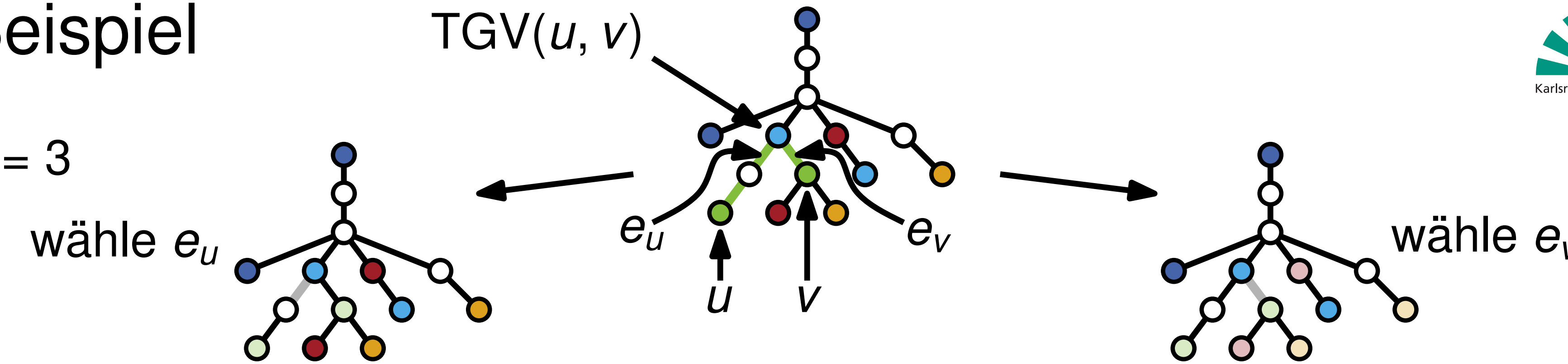
Beispiel

$k = 3$



Beispiel

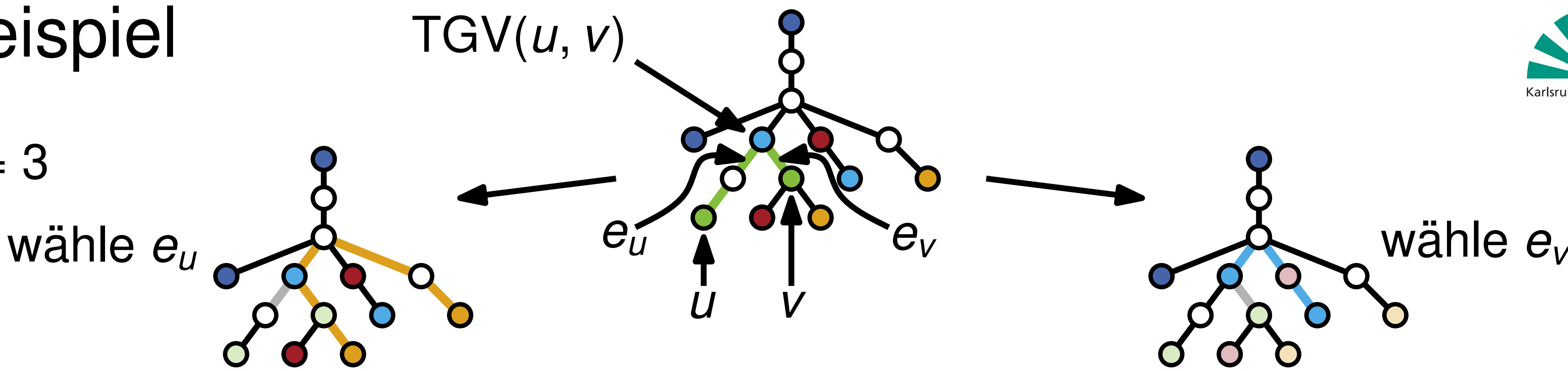
$k = 3$



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

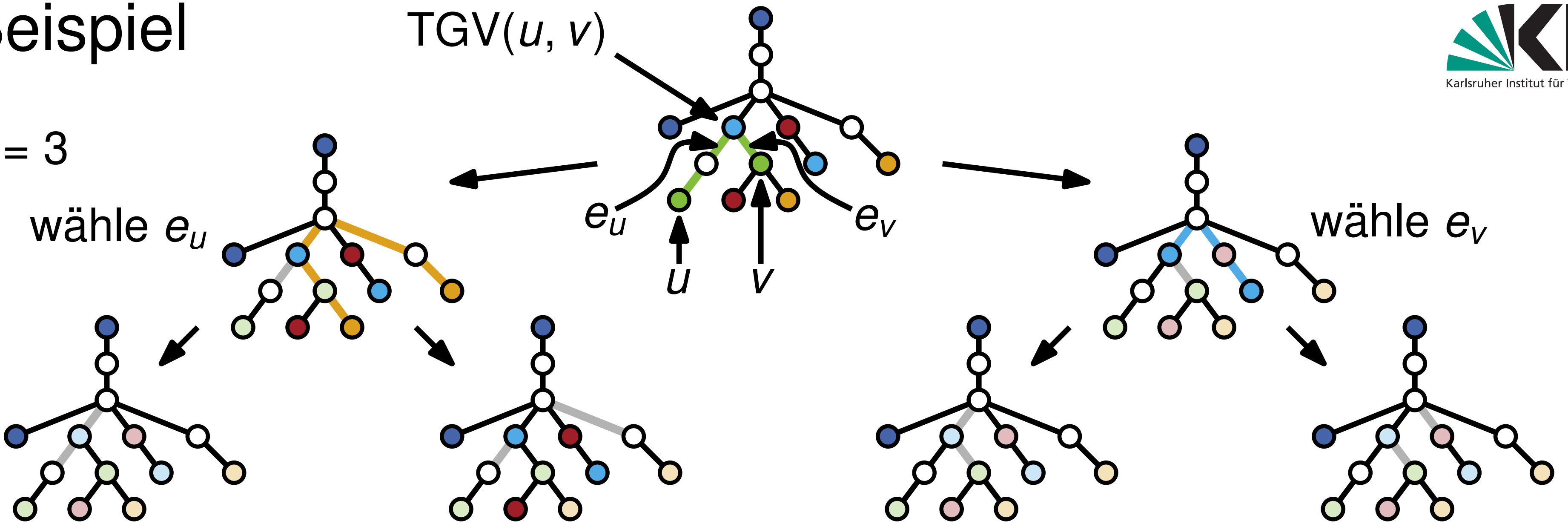
Beispiel

$k = 3$



Beispiel

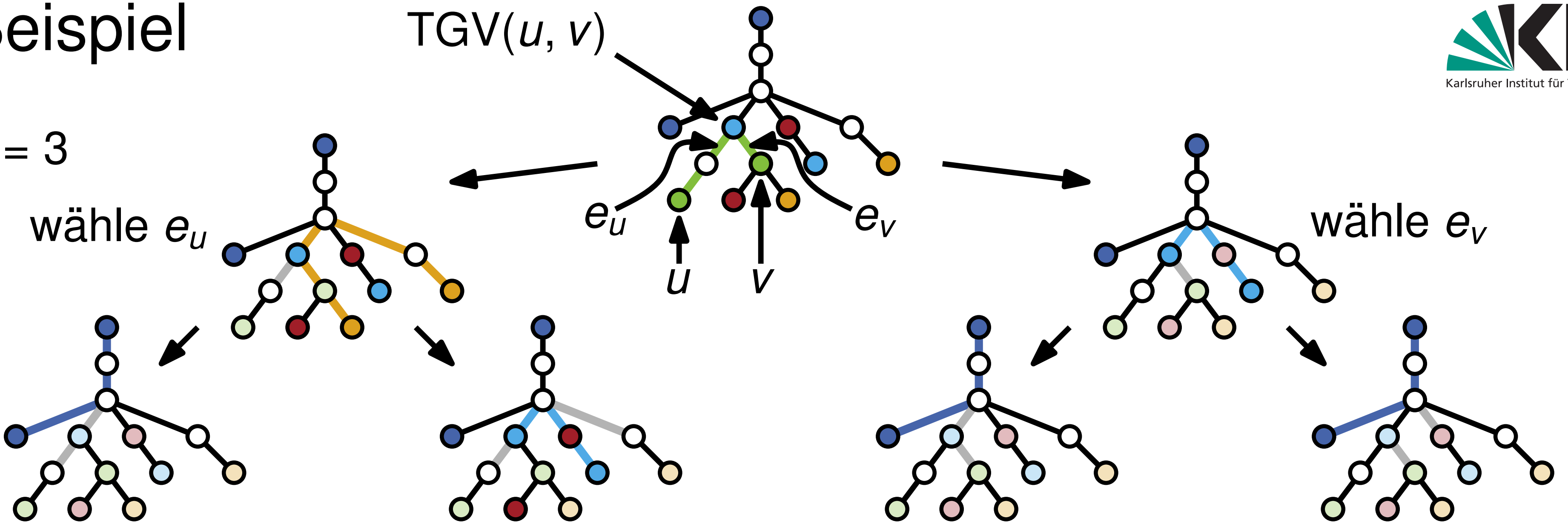
$k = 3$



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Beispiel

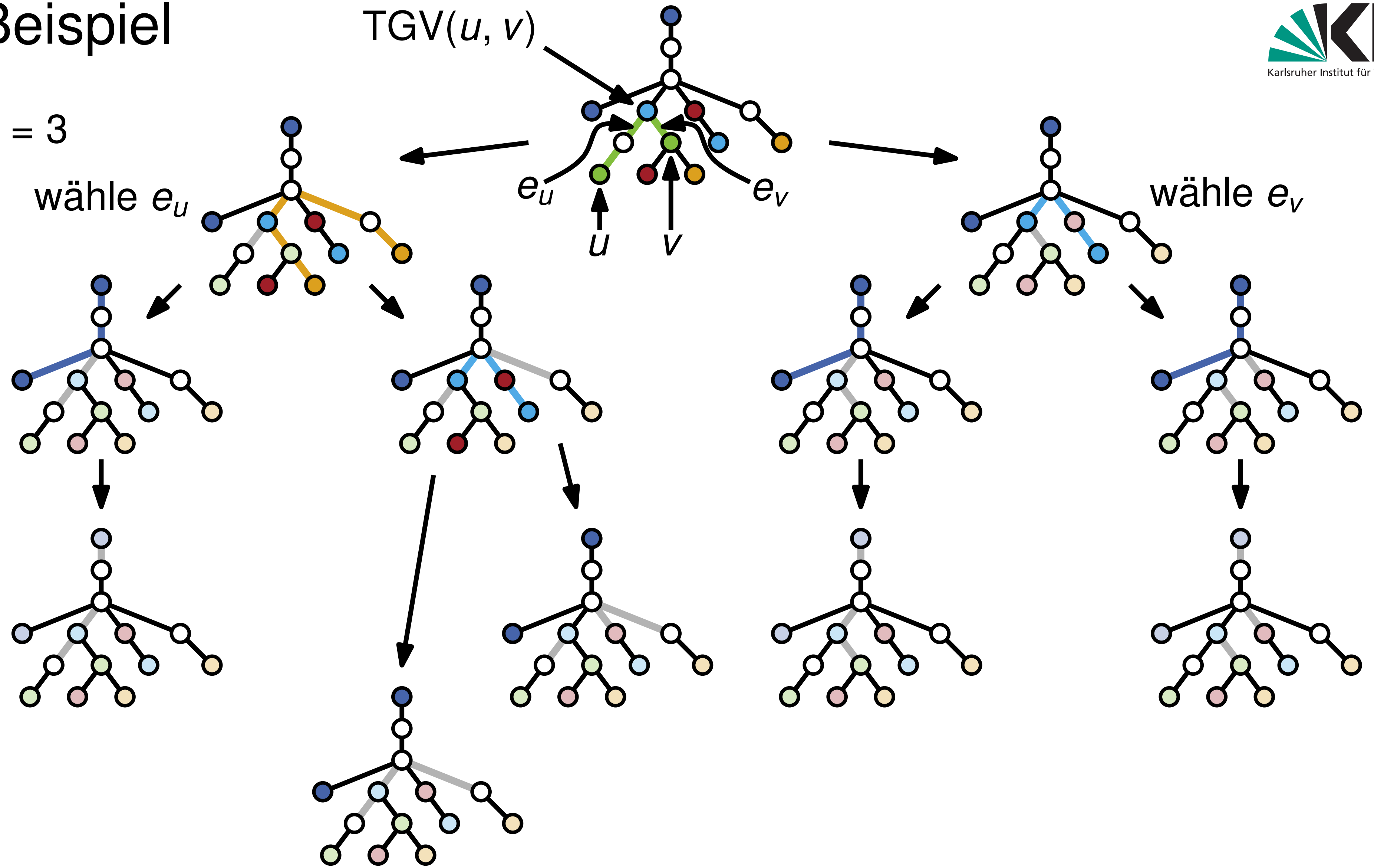
$k = 3$



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

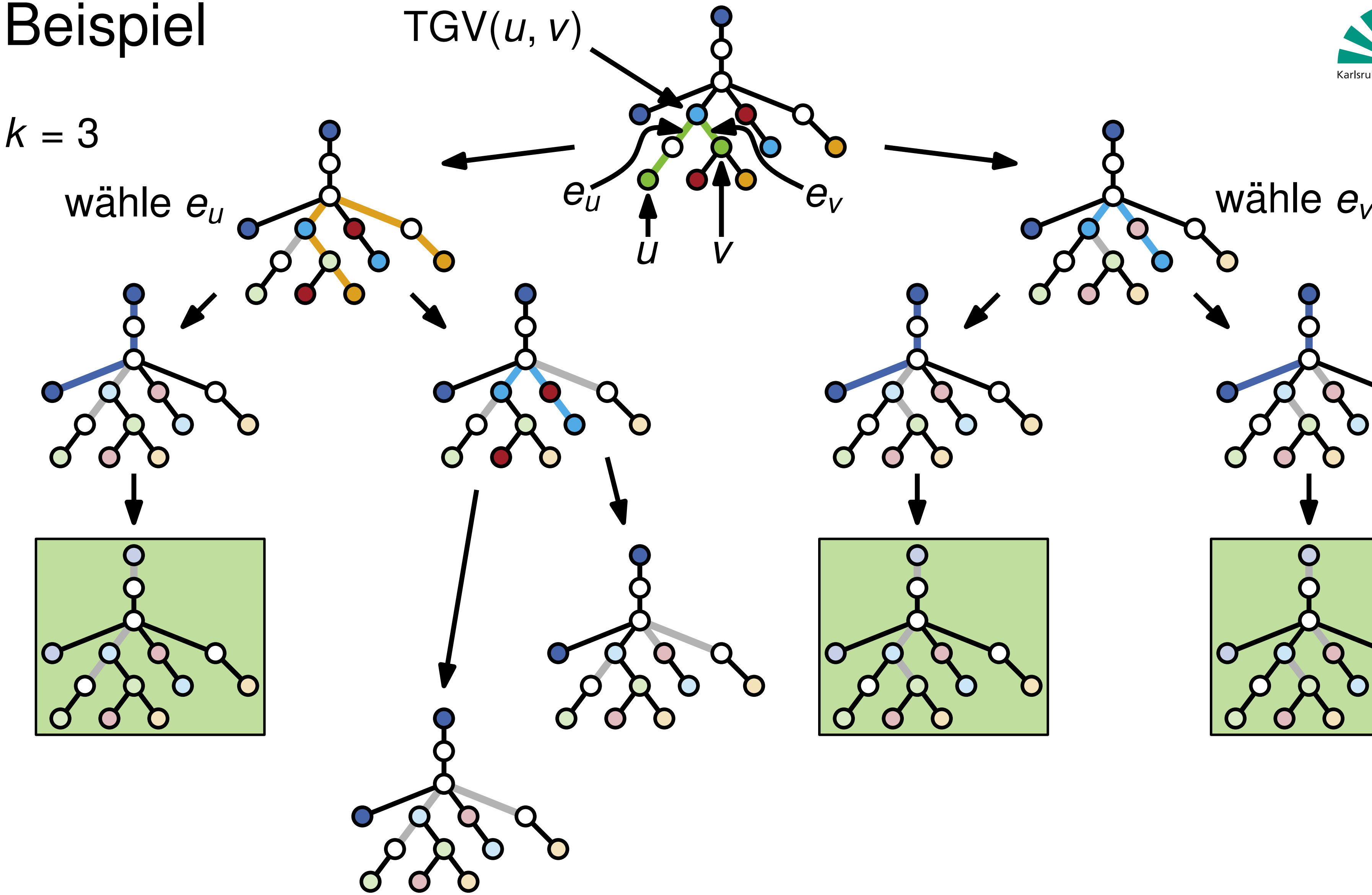
Beispiel

$k = 3$

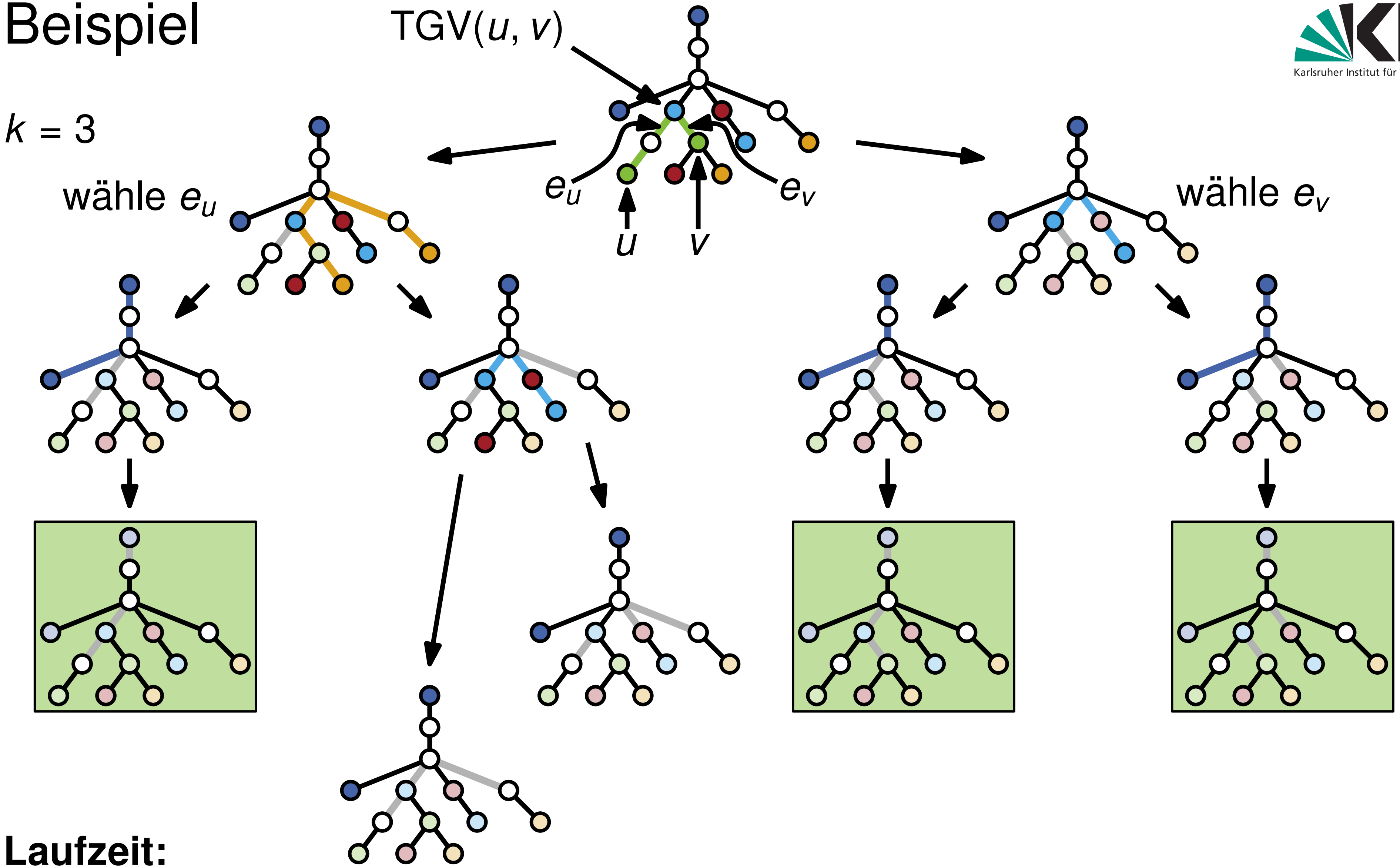


Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Beispiel



Beispiel



Laufzeit:

- Binärbaum der Tiefe $k \Rightarrow O(2^k)$ viele Schritte.
- Laufzeit $O(n^2)$ pro Schritt. $\Rightarrow$ Gesamtlaufzeit $O(2^k \cdot n^2)$