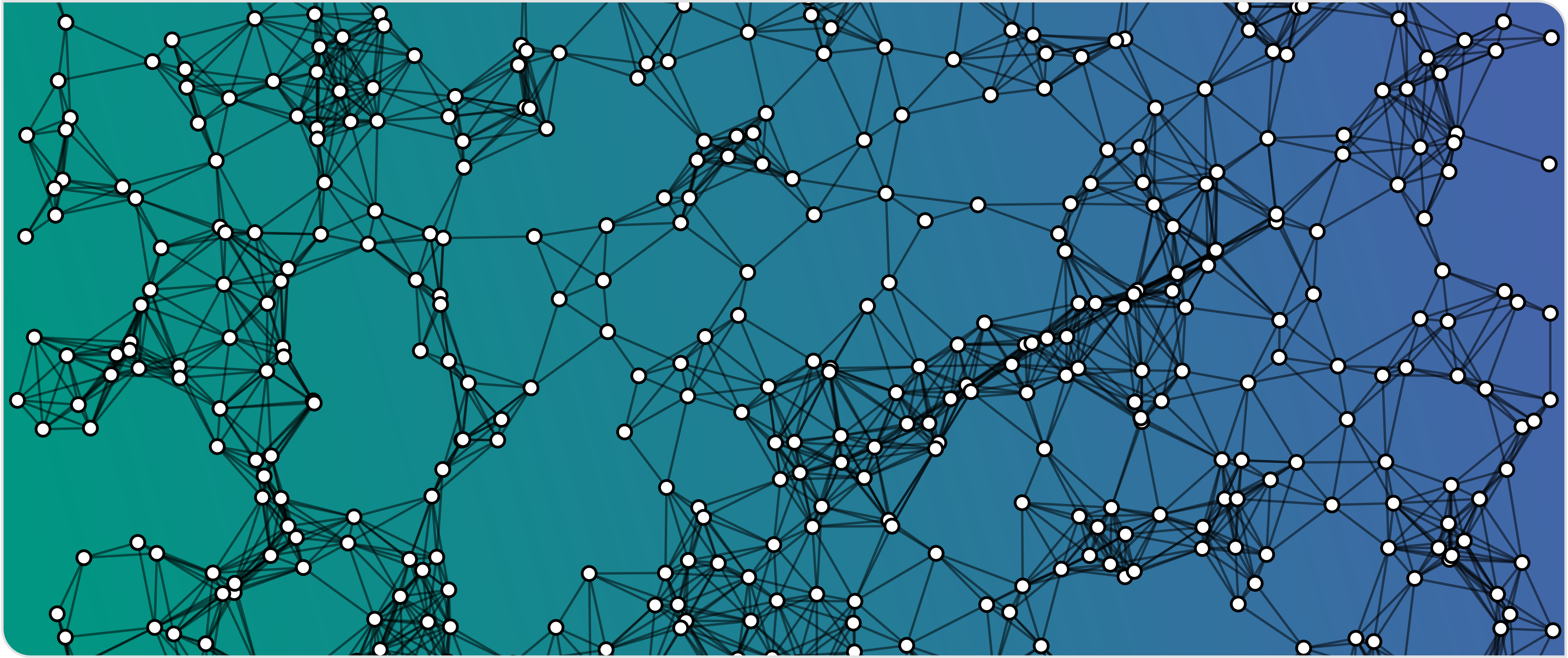
Karlsruher Institut für Technologie

Parametrisierte Algorithmen

Übung 6



Heute

Übungsblatt 5

Übungsblatt 6

Heute

Übungsblatt 5

- 5-Sterne
- LONGEST CYCLE
- DOMINATING SET auf speziellen Graphen
- Coding: CLOSEST-STRING

Übungsblatt 6

Heute

Übungsblatt 5

- 5-Sterne
- LONGEST CYCLE
- DOMINATING SET auf speziellen Graphen
- Coding: CLOSEST-STRING

Übungsblatt 6

- Verschiedenes zu Baumweite
- dominierende Mengen
- Baumweite planarer Graphen
- Coding: DP auf Baumzerlegung

Heute

Übungsblatt 5

- 5-Sterne
- LONGEST CYCLE
- DOMINATING SET auf speziellen Graphen
- Coding: CLOSEST-STRING

Übungsblatt 6

- Verschiedenes zu Baumweite
- dominierende Mengen
- Baumweite planarer Graphen
- Coding: DP auf Baumzerlegung

Außerdem?

Heute

Übungsblatt 5

- 5-Sterne
- LONGEST CYCLE
- DOMINATING SET auf speziellen Graphen
- Coding: CLOSEST-STRING

Übungsblatt 6

- Verschiedenes zu Baumweite
- dominierende Mengen
- Baumweite planarer Graphen
- Coding: DP auf Baumzerlegung

Außerdem?

- Wie rechnet man Baumweite aus?

Heute

Übungsblatt 5

- 5-Sterne
- LONGEST CYCLE
- DOMINATING SET auf speziellen Graphen
- Coding: CLOSEST-STRING

Übungsblatt 6

- Verschiedenes zu Baumweite
- dominierende Mengen
- Baumweite planarer Graphen
- Coding: DP auf Baumzerlegung

Außerdem?

- Wie rechnet man Baumweite aus? (in der echten Welt)

Heute

Übungsblatt 5

- 5-Sterne
- Longest Cycle
- Dominating Set auf speziellen Graphen
- Coding: Closest-String

Übungsblatt 6

- Verschiedenes zu Baumweite
- dominierende Mengen
- Baumweite planarer Graphen
- Coding: DP auf Baumzerlegung

Außerdem?

- Wie rechnet man Baumweite aus? (in der echten Welt)
- noch eine Charakterisierung von Baumweite

Heute

Übungsblatt 5

- 5-Sterne
- LONGEST CYCLE
- DOMINATING SET auf speziellen Graphen
- Coding: CLOSEST-STRING

Übungsblatt 6

- Verschiedenes zu Baumweite
- dominierende Mengen
- Baumweite planarer Graphen
- Coding: DP auf Baumzerlegung

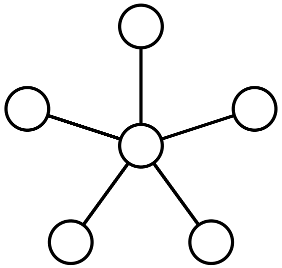
Außerdem?

- Wie rechnet man Baumweite aus? (in der echten Welt)
- noch eine Charakterisierung von Baumweite
- Evaluation

Übungsblatt 5: Sterne

Gegeben: Gegeben Graph $G = (V, E)$ und Parameter k

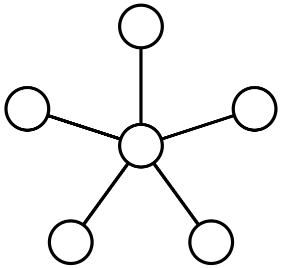
Frage: enthält G mindestens k knotendisjunkte induzierte 5-Sterne?



Übungsblatt 5: Sterne

Gegeben: Gegeben Graph $G = (V, E)$ und Parameter k

Frage: enthält G mindestens k knotendisjunkte induzierte 5-Sterne?

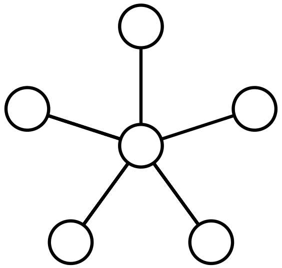


Idee: Verwende Color-Coding

Übungsblatt 5: Sterne

Gegeben: Gegeben Graph $G = (V, E)$ und Parameter k

Frage: enthält G mindestens k knotendisjunkte induzierte 5-Sterne?



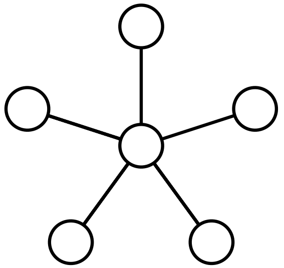
Idee: Verwende Color-Coding

- Reminder: (n, k) perfekte Familie von Hash-Funktionen:
  - Menge von Färbungen der n Elemente (Knoten) mit k Farben
  - jede k-elementige Teilmenge ist in mindestens einer Färbung bunt
  - nennt man auch: (n, k) perfekte Familie von Hash-Funktionen
  - $O(6.4^k \log^2 n)$ Funktionen, Konstruktion in $O(6.4^k n \log^2 n)$ Zeit

Übungsblatt 5: Sterne

Gegeben: Gegeben Graph $G = (V, E)$ und Parameter k

Frage: enthält G mindestens k knotendisjunkte induzierte 5-Sterne?



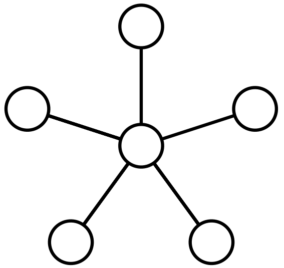
Idee: Verwende Color-Coding

- Reminder: (n, k) perfekte Familie von Hash-Funktionen:
  - Menge von Färbungen der n Elemente (Knoten) mit k Farben
  - jede k-elementige Teilmenge ist in mindestens einer Färbung bunt
  - nennt man auch: (n, k) perfekte Familie von Hash-Funktionen
  - $O(6.4^k \log^2 n)$ Funktionen, Konstruktion in $O(6.4^k n \log^2 n)$ Zeit

Verwendung hier:

Übungsblatt 5: Sterne

Gegeben: Gegeben Graph $G = (V, E)$ und Parameter k



Frage: enthält G mindestens k knotendisjunkte induzierte 5-Sterne?

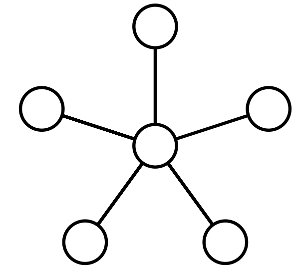
Idee: Verwende Color-Coding

- Reminder: (n, k) perfekte Familie von Hash-Funktionen:
  - Menge von Färbungen der n Elemente (Knoten) mit k Farben
  - jede k-elementige Teilmenge ist in mindestens einer Färbung bunt
  - nennt man auch: (n, k) perfekte Familie von Hash-Funktionen
  - $O(6.4^k \log^2 n)$ Funktionen, Konstruktion in $O(6.4^k n \log^2 n)$ Zeit

Verwendung hier: Betrachte $(n, 6k)$ perf. FvH-F

Übungsblatt 5: Sterne

Gegeben: Gegeben Graph $G = (V, E)$ und Parameter k

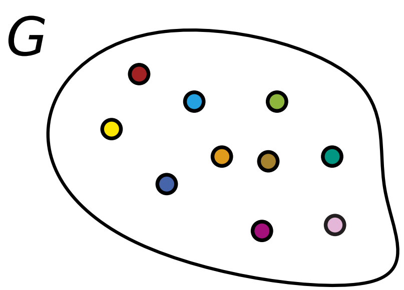


Frage: enthält G mindestens k knotendisjunkte induzierte 5-Sterne?

Idee: Verwende Color-Coding

- Reminder: (n, k) perfekte Familie von Hash-Funktionen:
  - Menge von Färbungen der n Elemente (Knoten) mit k Farben
  - jede k-elementige Teilmenge ist in mindestens einer Färbung bunt
  - nennt man auch: (n, k) perfekte Familie von Hash-Funktionen
  - $O(6.4^k \log^2 n)$ Funktionen, Konstruktion in $O(6.4^k n \log^2 n)$ Zeit

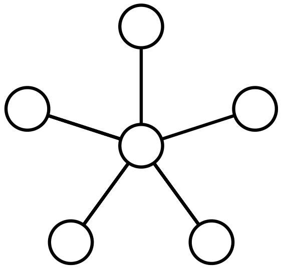
Verwendung hier: Betrachte $(n, 6k)$ perf. FvH-F



Übungsblatt 5: Sterne

Gegeben: Gegeben Graph $G = (V, E)$ und Parameter k

Frage: enthält G mindestens k knotendisjunkte induzierte 5-Sterne?

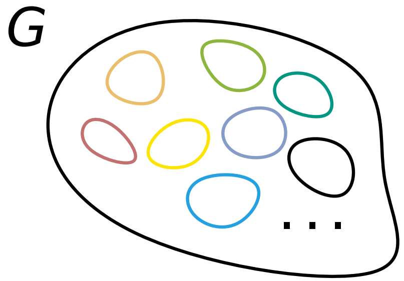


Idee: Verwende Color-Coding

- Reminder: (n, k) perfekte Familie von Hash-Funktionen:
  - Menge von Färbungen der n Elemente (Knoten) mit k Farben
  - jede k-elementige Teilmenge ist in mindestens einer Färbung bunt
  - nennt man auch: (n, k) perfekte Familie von Hash-Funktionen
  - $O(6.4^k \log^2 n)$ Funktionen, Konstruktion in $O(6.4^k n \log^2 n)$ Zeit

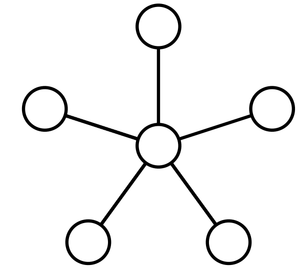
Verwendung hier: Betrachte $(n, 6k)$ perf. FvH-F

- $6k$ Farben in k Gruppen einteilen



Übungsblatt 5: Sterne

Gegeben: Gegeben Graph $G = (V, E)$ und Parameter k

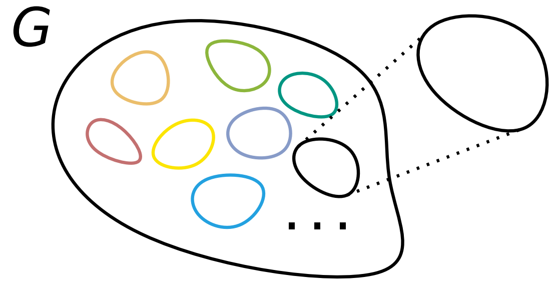


Frage: enthält G mindestens k knotendisjunkte induzierte 5-Sterne?

Idee: Verwende Color-Coding

- Reminder: (n, k) perfekte Familie von Hash-Funktionen:
  - Menge von Färbungen der n Elemente (Knoten) mit k Farben
  - jede k-elementige Teilmenge ist in mindestens einer Färbung bunt
  - nennt man auch: (n, k) perfekte Familie von Hash-Funktionen
  - $O(6.4^k \log^2 n)$ Funktionen, Konstruktion in $O(6.4^k n \log^2 n)$ Zeit

Verwendung hier: Betrachte $(n, 6k)$ perf. FvH-F

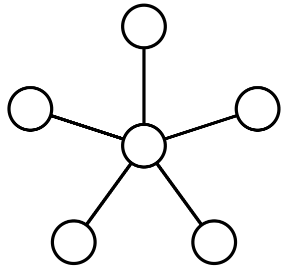


- $6k$ Farben in k Gruppen einteilen

Übungsblatt 5: Sterne

Gegeben: Gegeben Graph $G = (V, E)$ und Parameter k

Frage: enthält G mindestens k knotendisjunkte induzierte 5-Sterne?

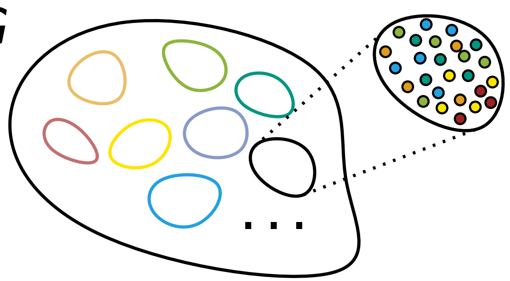


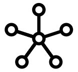
Idee: Verwende Color-Coding

- Reminder: (n, k) perfekte Familie von Hash-Funktionen:
  - Menge von Färbungen der n Elemente (Knoten) mit k Farben
  - jede k-elementige Teilmenge ist in mindestens einer Färbung bunt
  - nennt man auch: (n, k) perfekte Familie von Hash-Funktionen
  - $O(6.4^k \log^2 n)$ Funktionen, Konstruktion in $O(6.4^k n \log^2 n)$ Zeit

Verwendung hier: Betrachte $(n, 6k)$ perf. FvH-F $O((6.4)^k n \log^2 n)$

G



- $6k$ Farben in k Gruppen einteilen $O((6k)!)$
- Überprüfe alle 6-Tupel: bunt,  $O(n^6)$
- bei Ja-Instanz: jede Gruppe hat einen bunten 5-Stern

Übungsblatt 5: Longest Cycle

Gegeben: Gegeben Graph $G = (V, E)$ und Parameter k

Frage: enthält G einen Kreis der Länge mindestenz k?

Übungsblatt 5: LONGEST CYCLE

Gegeben: Gegeben Graph $G = (V, E)$ und Parameter k

Frage: enthält G einen Kreis der Länge mindestenz k?

Aus der Vorlesung bekannt: Colorful LONGEST PATH
Gegeben ein Graph $G = (V, E)$, einen Parameter k, sowie eine Partition (Färbung) $V_1, \ldots, V_k$ von V. Gibt es einen bunten Pfad der Länge k in G?
(Colorful LONGEST PATH kann in $O(2^k km)$ Zeit gelöst werden.)

Übungsblatt 5: Longest Cycle

Gegeben: Gegeben Graph $G = (V, E)$ und Parameter k

Frage: enthält G einen Kreis der Länge mindestenz k?

Aus der Vorlesung bekannt: Colorful Longest Path
Gegeben ein Graph $G = (V, E)$, einen Parameter k, sowie eine Partition (Färbung) $V_1, \dots, V_k$ von V. Gibt es einen bunten Pfad der Länge k in G?
(Colorful Longest Path kann in $O(2^k km)$ Zeit gelöst werden.)

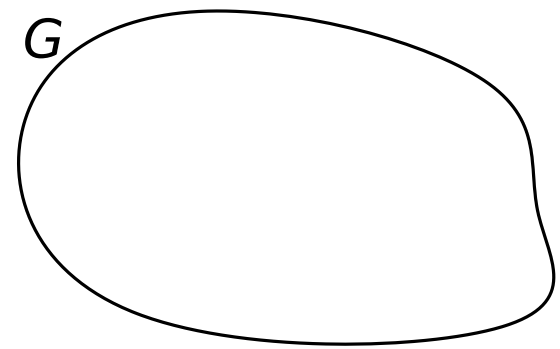
Übungsblatt 5: Longest Cycle

Gegeben: Gegeben Graph $G = (V, E)$ und Parameter k

Frage: enthält G einen Kreis der Länge mindestenz k?

Aus der Vorlesung bekannt: Colorful Longest Path
Gegeben ein Graph $G = (V, E)$, einen Parameter k, sowie eine Partition (Färbung) $V_1, \dots, V_k$ von V. Gibt es einen bunten Pfad der Länge k in G?
(Colorful Longest Path kann in $O(2^k km)$ Zeit gelöst werden.)

Idee 1: Kreise der Länge genau k finden



Übungsblatt 5: Longest Cycle

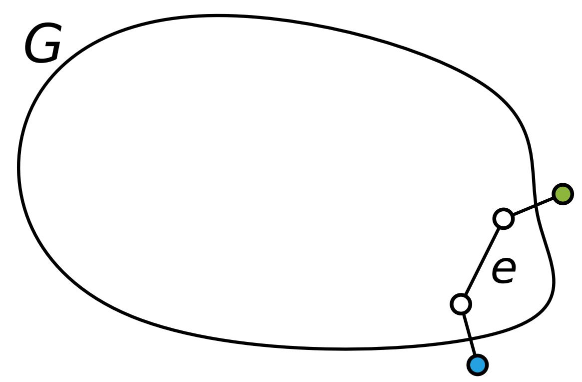
Gegeben: Gegeben Graph $G = (V, E)$ und Parameter k

Frage: enthält G einen Kreis der Länge mindestenz k?

Aus der Vorlesung bekannt: Colorful Longest Path
Gegeben ein Graph $G = (V, E)$, einen Parameter k, sowie eine Partition (Färbung) $V_1, \dots, V_k$ von V. Gibt es einen bunten Pfad der Länge k in G?
(Colorful Longest Path kann in $O(2^k km)$ Zeit gelöst werden.)

Idee 1: Kreise der Länge genau k finden

- neue Blätter u' und v' mit neuen Farben an Kante $e = \{u, v\}$



Übungsblatt 5: Longest Cycle

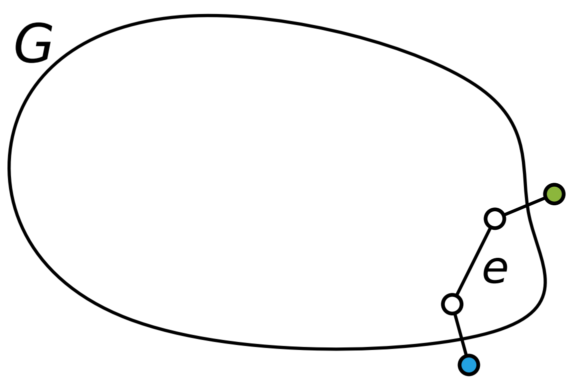
Gegeben: Gegeben Graph $G = (V, E)$ und Parameter k

Frage: enthält G einen Kreis der Länge mindestenz k?

Aus der Vorlesung bekannt: Colorful Longest Path
Gegeben ein Graph $G = (V, E)$, einen Parameter k, sowie eine Partition (Färbung) $V_1, \dots, V_k$ von V. Gibt es einen bunten Pfad der Länge k in G?
(Colorful Longest Path kann in $O(2^k km)$ Zeit gelöst werden.)

Idee 1: Kreise der Länge genau k finden

- neue Blätter u' und v' mit neuen Farben an Kante $e = \{u, v\}$
- suche bunten Pfad der Länge $k + 2$



Übungsblatt 5: Longest Cycle

Gegeben: Gegeben Graph $G = (V, E)$ und Parameter k

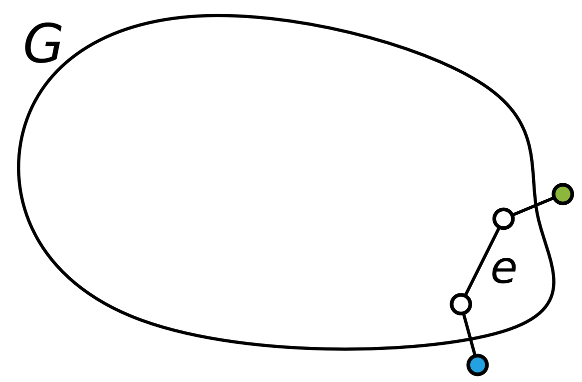
Frage: enthält G einen Kreis der Länge mindestenz k?

Aus der Vorlesung bekannt: Colorful Longest Path
Gegeben ein Graph $G = (V, E)$, einen Parameter k, sowie eine Partition (Färbung) $V_1, \dots, V_k$ von V. Gibt es einen bunten Pfad der Länge k in G?

(Colorful Longest Path kann in $O(2^k km)$ Zeit gelöst werden.)

Idee 1: Kreise der Länge genau k finden

- neue Blätter u' und v' mit neuen Farben an Kante $e = \{u, v\}$
- suche bunten Pfad der Länge $k + 2$



Idee 2: Kreise iterativ schrumpfen

Übungsblatt 5: Longest Cycle

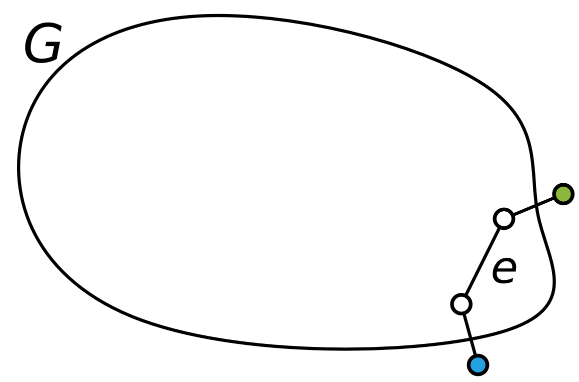
Gegeben: Gegeben Graph $G = (V, E)$ und Parameter k

Frage: enthält G einen Kreis der Länge mindestenz k?

Aus der Vorlesung bekannt: Colorful Longest Path
Gegeben ein Graph $G = (V, E)$, einen Parameter k, sowie eine Partition (Färbung) $V_1, \ldots, V_k$ von V. Gibt es einen bunten Pfad der Länge k in G?
(Colorful Longest Path kann in $O(2^k km)$ Zeit gelöst werden.)

Idee 1: Kreise der Länge genau k finden

- neue Blätter u' und v' mit neuen Farben an Kante $e = \{u, v\}$
- suche bunten Pfad der Länge $k + 2$



Idee 2: Kreise iterativ schrumpfen

- Kantenkontraktion: Kreis bleibt, wird halbiert oder eine Kante kürzer

Übungsblatt 5: Longest Cycle

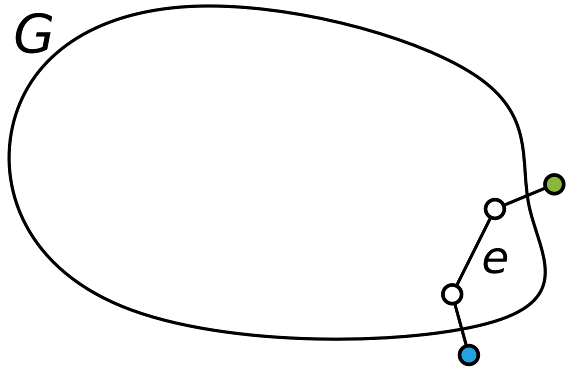
Gegeben: Gegeben Graph $G = (V, E)$ und Parameter k

Frage: enthält G einen Kreis der Länge mindestenz k?

Aus der Vorlesung bekannt: Colorful Longest Path
Gegeben ein Graph $G = (V, E)$, einen Parameter k, sowie eine Partition (Färbung) $V_1, \dots, V_k$ von V. Gibt es einen bunten Pfad der Länge k in G?
(Colorful Longest Path kann in $O(2^k km)$ Zeit gelöst werden.)

Idee 1: Kreise der Länge genau k finden

- neue Blätter u' und v' mit neuen Farben an Kante $e = \{u, v\}$
- suche bunten Pfad der Länge $k + 2$



Idee 2: Kreise iterativ schrumpfen

- Kantenkontraktion: Kreis bleibt, wird halbiert oder eine Kante kürzer
- suche Kreise der Länge k bis $2k$

Übungsblatt 5: Longest Cycle

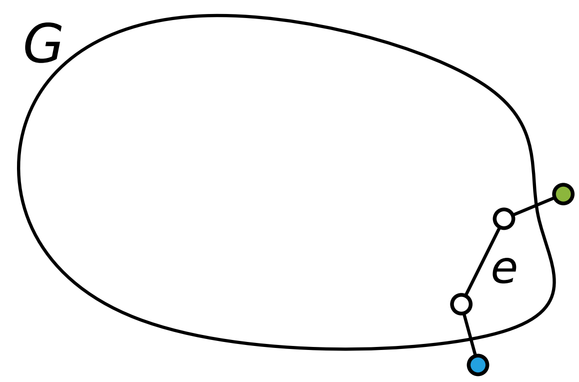
Gegeben: Gegeben Graph $G = (V, E)$ und Parameter k

Frage: enthält G einen Kreis der Länge mindestenz k?

Aus der Vorlesung bekannt: Colorful Longest Path
Gegeben ein Graph $G = (V, E)$, einen Parameter k, sowie eine Partition (Färbung) $V_1, \dots, V_k$ von V. Gibt es einen bunten Pfad der Länge k in G?
(Colorful Longest Path kann in $O(2^k km)$ Zeit gelöst werden.)

Idee 1: Kreise der Länge genau k finden

- neue Blätter u' und v' mit neuen Farben an Kante $e = \{u, v\}$
- suche bunten Pfad der Länge $k + 2$



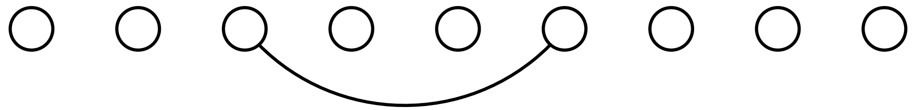
Idee 2: Kreise iterativ schrumpfen

- Kantenkontraktion: Kreis bleibt, wird halbiert oder eine Kante kürzer
- suche Kreise der Länge k bis $2k$

iterieren

Übungsblatt 5: Dominating Set

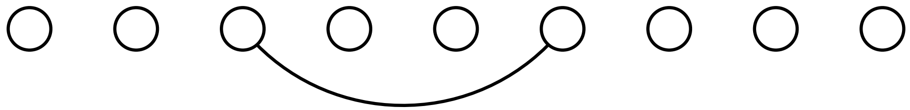
Gegeben: Gegeben Graph G mit Knotenordnung $V = \{v_1, \ldots, v_n\}$ und Parameter k



maximale Kantenlänge:
$\{v_i, v_j\} \in E \Rightarrow |i - j| \leq k$

Übungsblatt 5: Dominating Set

Gegeben: Gegeben Graph G mit Knotenordnung $V = \{v_1, \dots, v_n\}$ und Parameter k

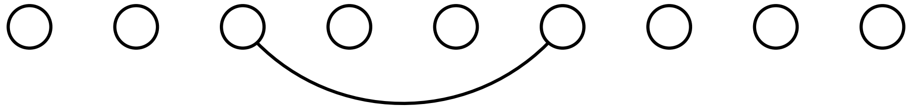


maximale Kantenlänge:
$\{v_i, v_j\} \in E \Rightarrow |i - j| \leq k$

Aufgabe: berechne maximales Dominating Set

Übungsblatt 5: Dominating Set

Gegeben: Gegeben Graph G mit Knotenordnung $V = \{v_1, \dots, v_n\}$ und Parameter k



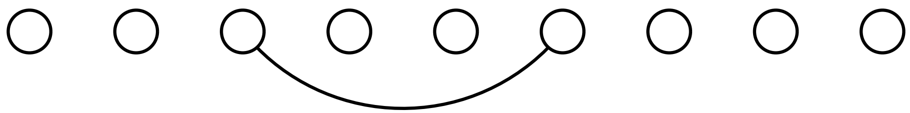
maximale Kantenlänge:
$\{v_i, v_j\} \in E \Rightarrow |i - j| \leq k$

Aufgabe: berechne maximales Dominating Set

Idee: dynamisches Programm auf schöner Pfadzerlegung

Übungsblatt 5: Dominating Set

Gegeben: Gegeben Graph G mit Knotenordnung $V = \{v_1, \ldots, v_n\}$ und Parameter k



maximale Kantenlänge:
$\{v_i, v_j\} \in E \Rightarrow |i - j| \leq k$

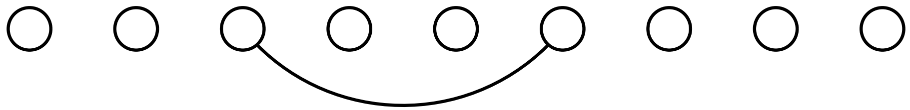
Aufgabe: berechne maximales Dominating Set

Idee: dynamisches Programm auf schöner Pfadzerlegung

- Zutaten für DP auf Pfad/Baum-Zerlegung
  1. Bedeutung der Teillösungen
  2. Berechnung der Teillösungen (*introduce*, *forget*, ...
  3. Weitere Details: Ablesen der Lösung

Übungsblatt 5: Dominating Set

Gegeben: Gegeben Graph G mit Knotenordnung $V = \{v_1, \dots, v_n\}$ und Parameter k



maximale Kantenlänge:
$\{v_i, v_j\} \in E \Rightarrow |i - j| \leq k$

Aufgabe: berechne maximales Dominating Set

Idee: dynamisches Programm auf schöner Pfadzerlegung

- Zutaten für DP auf Pfad/Baum-Zerlegung
  1. Bedeutung der Teillösungen
  2. Berechnung der Teillösungen (*introduce*, *forget*, ...
  3. Weitere Details: Ablesen der Lösung
- Außerdem: woher bekommen wir die Zerlegung?

Mehr zu Baumweite

Definition: (VL)
G chordal $\Leftrightarrow$ G hat keinen induzierten Kreis der Länge 4 oder mehr

Mehr zu Baumweite

Definition: (VL)
G chordal $\Leftrightarrow$ *G* hat keinen induzierten Kreis der Länge 4 oder mehr

Behauptung:
Baumweite von chordalen Graphen lässt sich in $n^{O(1)}$ entscheiden

Mehr zu Baumweite

Definition: (VL)
G chordal $\Leftrightarrow$ G hat keinen induzierten Kreis der Länge 4 oder mehr

Behauptung:
Baumweite von chordalen Graphen lässt sich in $n^{O(1)}$ entscheiden

Satz: Sei G chordal. Dann gibt es einen sogenannten *simplizialen* Knoten $v \in V(G)$ dessen Nachbarschaft eine Clique bildet.

Mehr zu Baumweite

Definition: (VL)
G chordal $\Leftrightarrow$ G hat keinen induzierten Kreis der Länge 4 oder mehr

Behauptung:
Baumweite von chordalen Graphen lässt sich in $n^{O(1)}$ entscheiden

Satz: Sei G chordal. Dann gibt es einen sogenannten *simplizialen* Knoten $v \in V(G)$ dessen Nachbarschaft eine Clique bildet. (Beweis auf Blatt 7)

Mehr zu Baumweite

Definition: (VL)
G chordal $\Leftrightarrow$ G hat keinen induzierten Kreis der Länge 4 oder mehr

Behauptung:
Baumweite von chordalen Graphen lässt sich in $n^{O(1)}$ entscheiden

Satz: Sei G chordal. Dann gibt es einen sogenannten *simplizialen* Knoten $v \in V(G)$ dessen Nachbarschaft eine Clique bildet. (Beweis auf Blatt 7)

Algorithmus:

Mehr zu Baumweite

Definition: (VL)
G chordal $\Leftrightarrow$ G hat keinen induzierten Kreis der Länge 4 oder mehr

Behauptung:
Baumweite von chordalen Graphen lässt sich in $n^{O(1)}$ entscheiden

Satz: Sei G chordal. Dann gibt es einen sogenannten *simplizialen* Knoten $v \in V(G)$ dessen Nachbarschaft eine Clique bildet. (Beweis auf Blatt 7)

Algorithmus:

- finde simplizialen Knoten v

Mehr zu Baumweite

Definition: (VL)
G chordal $\Leftrightarrow$ G hat keinen induzierten Kreis der Länge 4 oder mehr

Behauptung:
Baumweite von chordalen Graphen lässt sich in $n^{O(1)}$ entscheiden

Satz: Sei G chordal. Dann gibt es einen sogenannten *simplizialen* Knoten $v \in V(G)$ dessen Nachbarschaft eine Clique bildet. (Beweis auf Blatt 7)

Algorithmus:

- finde simplizialen Knoten v
- sei $(T, \{X_t\})$ Baumzerlegung von $G \setminus \{v\}$ (rekursiv)

Mehr zu Baumweite

Definition: (VL)
G chordal $\Leftrightarrow$ G hat keinen induzierten Kreis der Länge 4 oder mehr

Behauptung:
Baumweite von chordalen Graphen lässt sich in $n^{O(1)}$ entscheiden

Satz: Sei G chordal. Dann gibt es einen sogenannten *simplizialen* Knoten $v \in V(G)$ dessen Nachbarschaft eine Clique bildet. (Beweis auf Blatt 7)

Algorithmus:

- finde simplizialen Knoten v
- sei $(T, \{X_t\})$ Baumzerlegung von $G \setminus \{v\}$ (rekursiv)
- $(T, \{X_t\})$ mit zusätzlichem Blatt-Bag $N[v]$ verbunden zu Bag X_t mit $X_t \subseteq N(v)$ ist Baumzerlegung von G

Was wenn G nicht chordal ist?

Algorithmus:

- finde simplizialen Knoten v
- sei $(T, \{X_t\})$ Baumzerlegung von $G \setminus \{v\}$ (rekursiv)
- $(T, \{X_t\})$ mit zusätzlichem Blatt-Bag $N[v]$ verbunden zu Bag X_t mit $X_t \subseteq N(v)$ ist Baumzerlegung von G :)

Was wenn G nicht chordal ist?

Algorithmus:

- finde simplizialen Knoten v
- sei $(T, \{X_t\})$ Baumzerlegung von $G \setminus \{v\}$ (rekursiv)
- $(T, \{X_t\})$ mit zusätzlichem Blatt-Bag $N[v]$ verbunden zu Bag X_t mit $X_t \subseteq N(v)$ ist Baumzerlegung von G :)

Was wenn G nicht chordal ist?

Algorithmus:

- finde simplizialen Knoten v
- sei $(T, \{X_t\})$ Baumzerlegung von $G \setminus \{v\}$ (rekursiv)
- $(T, \{X_t\})$ mit zusätzlichem Blatt-Bag $N[v]$ verbunden zu Bag X_t mit $X_t \subseteq N(v)$ ist Baumzerlegung von G :)

Satz: Jeder Graph hat ein Eliminationsschema, mit dem sich die korrekte Baumweite bestimmen lässt.

Was wenn G nicht chordal ist?

Algorithmus:

- finde simplizialen Knoten v
- sei $(T, \{X_t\})$ Baumzerlegung von $G \setminus \{v\}$ (rekursiv)
- $(T, \{X_t\})$ mit zusätzlichem Blatt-Bag $N[v]$ verbunden zu Bag X_t mit $X_t \subseteq N(v)$ ist Baumzerlegung von G :)

Satz: Jeder Graph hat ein Eliminationsschema, mit dem sich die korrekte Baumweite bestimmen lässt.

Wieso?

Was wenn G nicht chordal ist?

Algorithmus:

- finde simplizialen Knoten v
- sei $(T, \{X_t\})$ Baumzerlegung von $G \setminus \{v\}$ (rekursiv)
- $(T, \{X_t\})$ mit zusätzlichem Blatt-Bag $N[v]$ verbunden zu Bag X_t mit $X_t \subseteq N(v)$ ist Baumzerlegung von G :)

Satz: Jeder Graph hat ein Eliminationsschema, mit dem sich die korrekte Baumweite bestimmen lässt.

Wieso?

Chordalweite

- $\text{chordalwidth}(G) = \min\{\omega(G') \mid G \subseteq G' \text{ und } G' \text{ ist chordal}\}$
- $\Rightarrow \text{tw}(G) = \text{chordalwidth}(G) - 1$

Was wenn G nicht chordal ist?

Karlsruher Institut für Technologie

Algorithmus:

- finde simplizialen Knoten v
- sei $(T, \{X_t\})$ Baumzerlegung von $G \setminus \{v\}$ (rekursiv)
- $(T, \{X_t\})$ mit zusätzlichem Blatt-Bag $N[v]$ verbunden zu Bag X_t mit $X_t \subseteq N(v)$ ist Baumzerlegung von G :)

Satz: Jeder Graph hat ein Eliminationsschema, mit dem sich die korrekte Baumweite bestimmen lässt.

Wieso?

Chordalweite

- $\text{chordalwidth}(G) = \min\{\omega(G') \mid G \subseteq G' \text{ und } G' \text{ ist chordal}\}$
- $\Rightarrow \text{tw}(G) = \text{chordalwidth}(G) - 1$

Baumweite lösen:

Was wenn G nicht chordal ist?

Algorithmus:

- finde simplizialen Knoten v
- sei $(T, \{X_t\})$ Baumzerlegung von $G \setminus \{v\}$ (rekursiv)
- $(T, \{X_t\})$ mit zusätzlichem Blatt-Bag $N[v]$ verbunden zu Bag X_t mit $X_t \subseteq N(v)$ ist Baumzerlegung von G :)

Satz: Jeder Graph hat ein Eliminationsschema, mit dem sich die korrekte Baumweite bestimmen lässt.

Wieso?

Chordalweite

- $\text{chordalwidth}(G) = \min\{\omega(G') \mid G \subseteq G' \text{ und } G' \text{ ist chordal}\}$
- $\Rightarrow \text{tw}(G) = \text{chordalwidth}(G) - 1$

Baumweite lösen:

- heuristisch: *min-degree* heuristic, *min-fill-in* heuristic

Was wenn G nicht chordal ist?

Algorithmus:

- finde simplizialen Knoten v
- sei $(T, \{X_t\})$ Baumzerlegung von $G \setminus \{v\}$ (rekursiv)
- $(T, \{X_t\})$ mit zusätzlichem Blatt-Bag $N[v]$ verbunden zu Bag X_t mit $X_t \subseteq N(v)$ ist Baumzerlegung von G :)

Satz: Jeder Graph hat ein Eliminationsschema, mit dem sich die korrekte Baumweite bestimmen lässt.

Wieso?

Chordalweite

- $\text{chordalwidth}(G) = \min\{\omega(G') \mid G \subseteq G' \text{ und } G' \text{ ist chordal}\}$
- $\Rightarrow \text{tw}(G) = \text{chordalwidth}(G) - 1$

Baumweite lösen:

- heuristisch: *min-degree* heuristic, *min-fill-in* heuristic
- heuristisch (fancy)

Was wenn G nicht chordal ist?

Algorithmus:

- finde simplizialen Knoten v
- sei $(T, \{X_t\})$ Baumzerlegung von $G \setminus \{v\}$ (rekursiv)
- $(T, \{X_t\})$ mit zusätzlichem Blatt-Bag $N[v]$ verbunden zu Bag X_t mit $X_t \subseteq N(v)$ ist Baumzerlegung von G :)

Satz: Jeder Graph hat ein Eliminationsschema, mit dem sich die korrekte Baumweite bestimmen lässt.

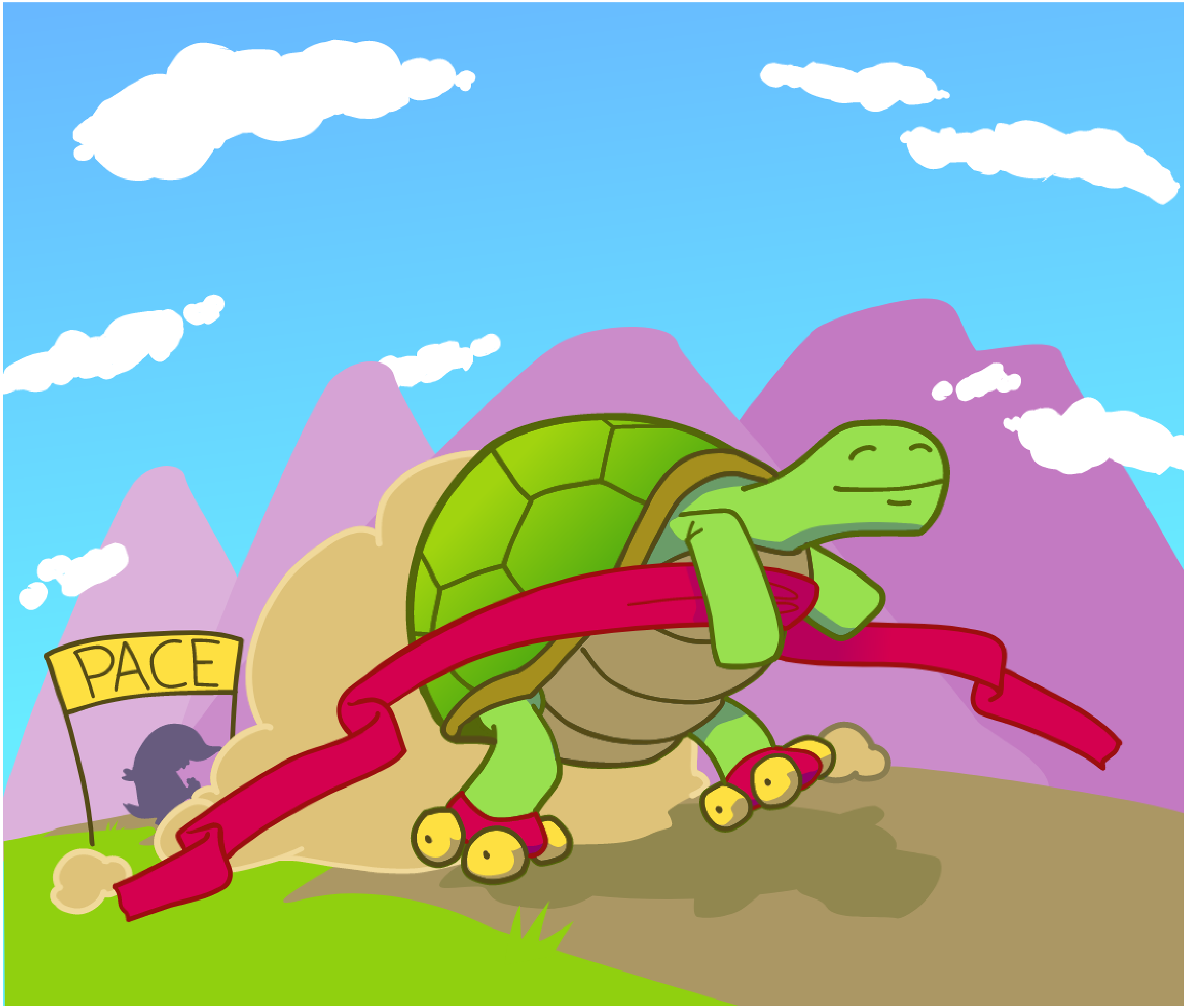
Wieso?

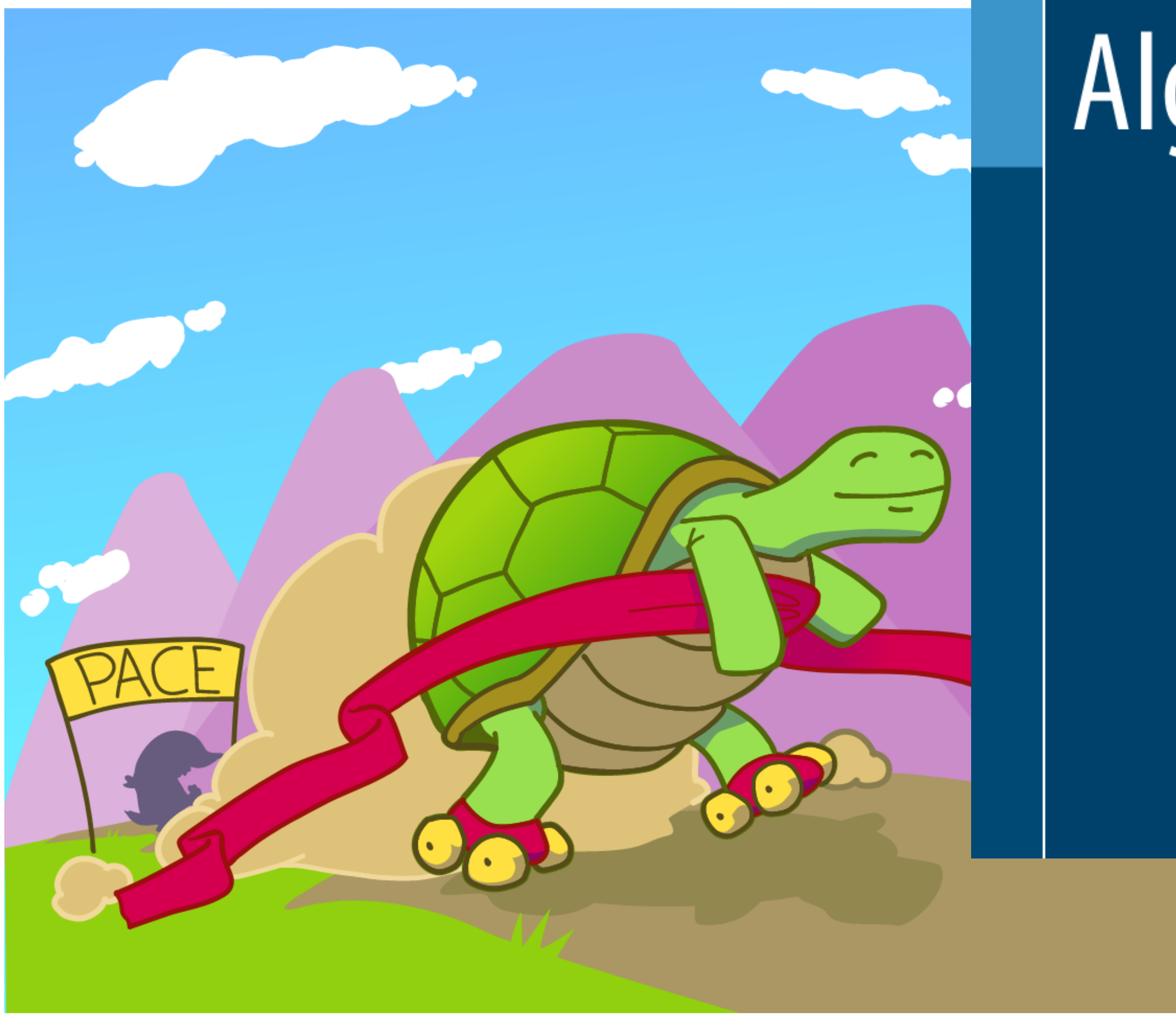
Chordalweite

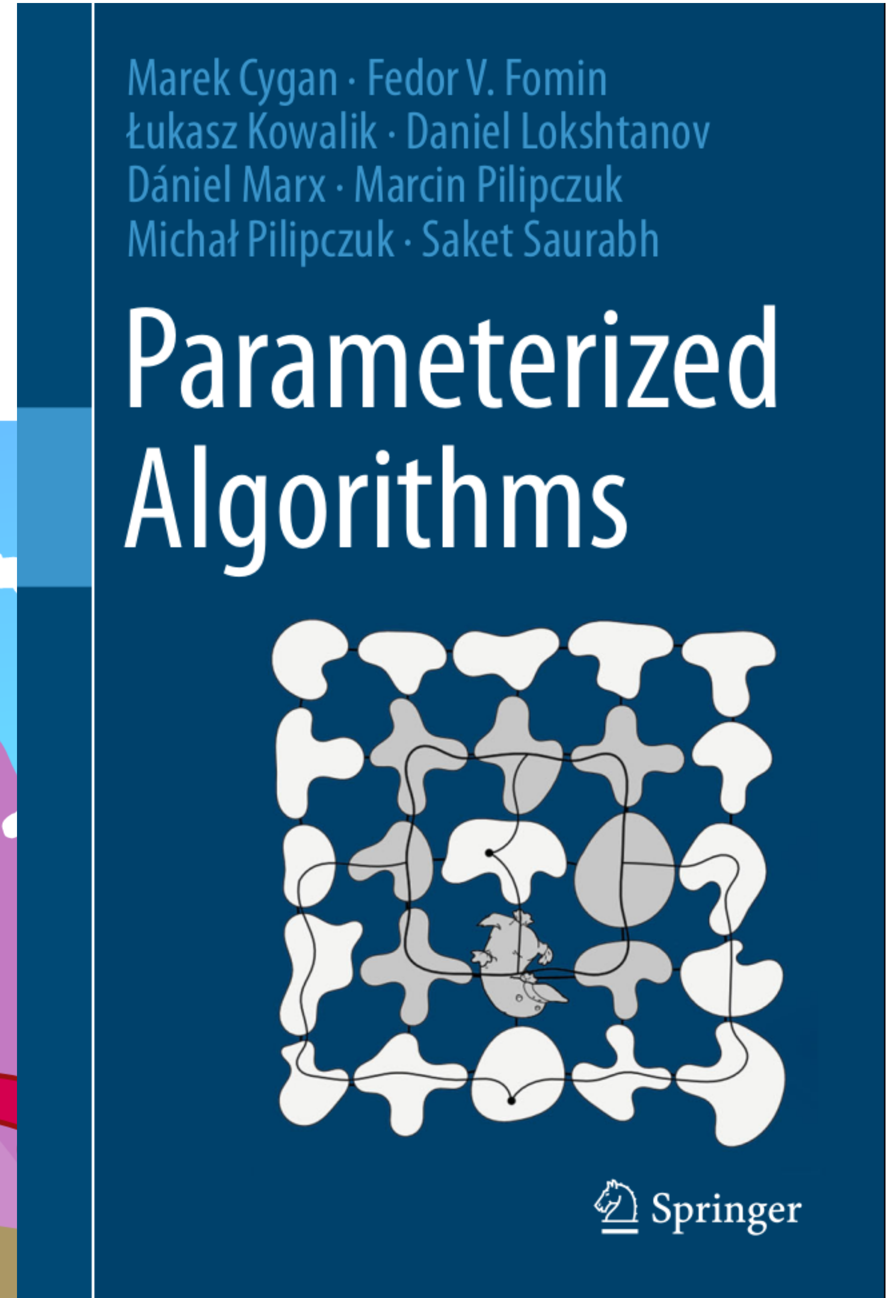
- chordalwidth$(G) = \min\{\omega(G') \mid G \subseteq G' \text{ und } G' \text{ ist chordal}\}$
- $\Rightarrow$ tw(G) = chordalwidth$(G) - 1$

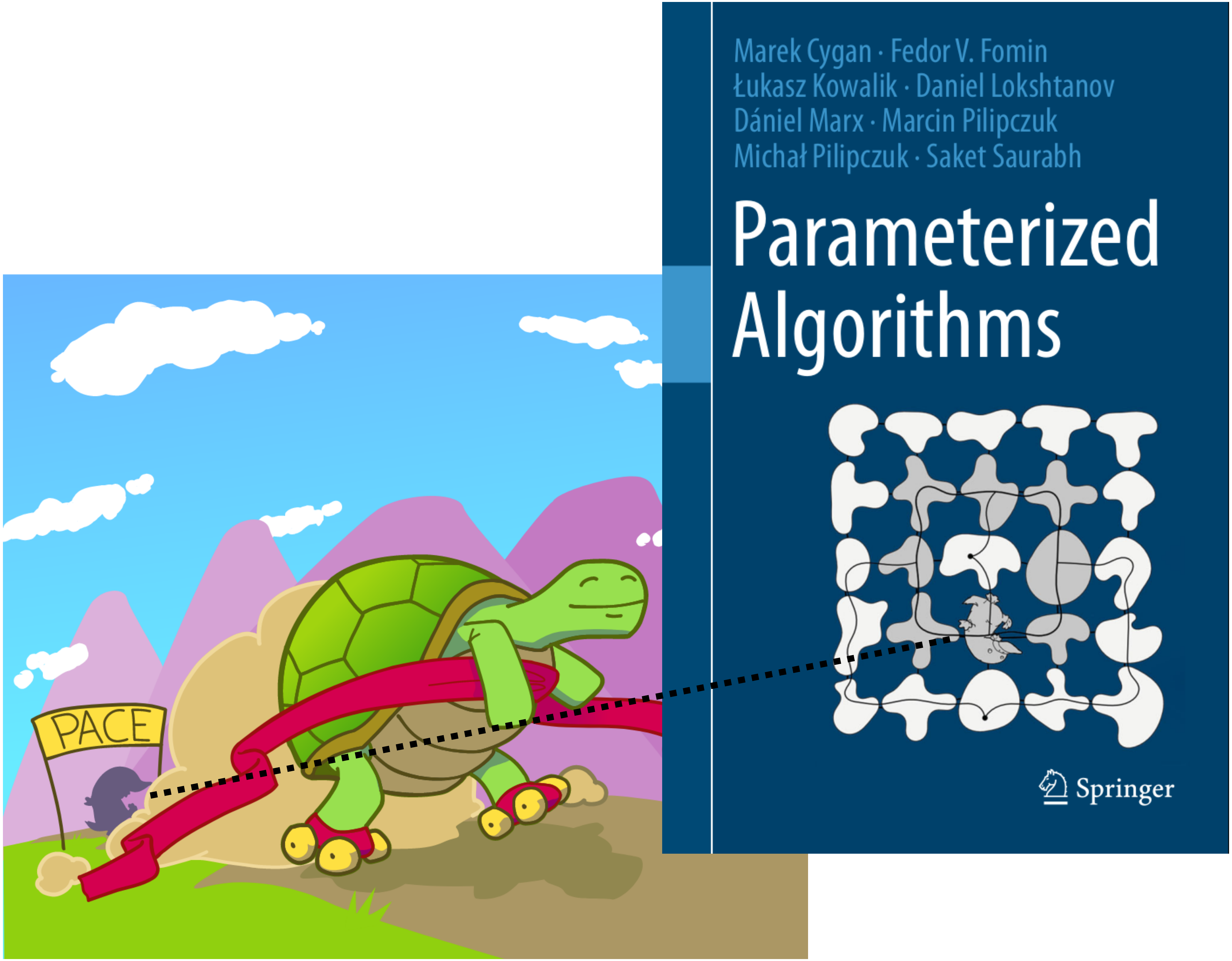
Baumweite lösen:

- heuristisch: *min-degree* heuristic, *min-fill-in* heuristic
- heuristisch (fancy)
- heuristisch (exakt)


PACE


PACE


Marek Cygan · Fedor V. Fomin
Łukasz Kowalik · Daniel Lokshtanov
Dániel Marx · Marcin Pilipczuk
Michał Pilipczuk · Saket Saurabh
Parameterized Algorithms
Springer


PACE
Marek Cygan · Fedor V. Fomin
Łukasz Kowalik · Daniel Lokshtanov
Dániel Marx · Marcin Pilipczuk
Michał Pilipczuk · Saket Saurabh
Parameterized Algorithms
Springer

Was wenn G nicht chordal ist?

Algorithmus:

- finde simplizialen Knoten v
- sei $(T, \{X_t\})$ Baumzerlegung von $G \setminus \{v\}$ (rekursiv)
- $(T, \{X_t\})$ mit zusätzlichem Blatt-Bag $N[v]$ verbunden zu Bag X_t mit $X_t \subseteq N(v)$ ist Baumzerlegung von G :)

Satz: Jeder Graph hat ein Eliminationsschema, mit dem sich die korrekte Baumweite bestimmen lässt.

Wieso?

Chordalweite

- $\text{chordalwidth}(G) = \min\{\omega(G') \mid G \subseteq G' \text{ und } G' \text{ ist chordal}\}$
- $\Rightarrow \text{tw}(G) = \text{chordalwidth}(G) - 1$

Baumweite lösen:

- heuristisch: *min-degree* heuristic, *min-fill-in* heuristic
- heuristisch (fancy)
- heuristisch (exakt)

Was wenn G nicht chordal ist?

Algorithmus:

- finde simplizialen Knoten v
- sei $(T, \{X_t\})$ Baumzerlegung von $G \setminus \{v\}$ (rekursiv)
- $(T, \{X_t\})$ mit zusätzlichem Blatt-Bag $N[v]$ verbunden zu Bag X_t mit $X_t \subseteq N(v)$ ist Baumzerlegung von G :)

Satz: Jeder Graph hat ein Eliminationsschema, mit dem sich die korrekte Baumweite bestimmen lässt.

Wieso?

Chordalweite
- $\text{chordalwidth}(G) = \min\{\omega(G') \mid G \subseteq G' \text{ und } G' \text{ ist chordal}\}$
- $\Rightarrow \text{tw}(G) = \text{chordalwidth}(G) - 1$

Baumweite lösen:

- heuristisch: *min-degree* heuristic, *min-fill-in* heuristic
- heuristisch (fancy)
- heuristisch (exakt)

$\left.\begin{array}{l}\text{heuristisch (fancy)}\\ \text{heuristisch (exakt)}\end{array}\right\}$ Paper aus PACE Challenge anschauen

Was wenn G nicht chordal ist?

Algorithmus:

- finde simplizialen Knoten v
- sei $(T, \{X_t\})$ Baumzerlegung von $G \setminus \{v\}$ (rekursiv)
- $(T, \{X_t\})$ mit zusätzlichem Blatt-Bag $N[v]$ verbunden zu Bag X_t mit $X_t \subseteq N(v)$ ist Baumzerlegung von G :)

Satz: Jeder Graph hat ein Eliminationsschema, mit dem sich die korrekte Baumweite bestimmen lässt.

Wieso?

Chordalweite

- $\text{chordalwidth}(G) = \min\{\omega(G') \mid G \subseteq G' \text{ und } G' \text{ ist chordal}\}$
- $\Rightarrow \text{tw}(G) = \text{chordalwidth}(G) - 1$

Baumweite lösen:

- heuristisch: *min-degree* heuristic, *min-fill-in* heuristic
- heuristisch (fancy)
- heuristisch (exakt)

Paper aus PACE Challenge anschauen

Positive-Instance Driven Dynamic Programming for Treewidth (ESA'17, best paper); Heuristic computation of exact treewidth (SEA'22); https://github.com/mabseher/htd

Cops and Robber

Spiel auf Graphen

- ein Bösewicht/Dieb
- mehrere Detektive / Polizisten
- Detektive wollen Dieb fangen

Spielaufbau

- Detektive stellen sich auf Knoten des Graphen, im Anschluss auch der Dieb

Rundenablauf

- ein Detektiv verlässt seinen Knoten, kündigt seine nächste Position an
- Dieb bewegt sich zu neuem Knoten
- der Detektiv stellt sich auf den angekündigten Knoten

Ziel

- Dieb verliert wenn er am Rundenende auf Knoten mit Polizisten steht

Cops and Robber

Spiel auf Graphen

- ein Bösewicht/Dieb
- mehrere Detektive / Polizisten
- Detektive wollen Dieb fangen

Zusammenhang zu Baumweite

- Was ist hier der Zusammenhang zu Baumweite?
- Wie genau dürfen sich Detektive und Bösewicht bewegen?

Spielaufbau

- Detektive stellen sich auf Knoten des Graphen, im Anschluss auch der Dieb

Rundenablauf

- ein Detektiv verlässt seinen Knoten, kündigt seine nächste Position an
- Dieb bewegt sich zu neuem Knoten
- der Detektiv stellt sich auf den angekündigten Knoten

Ziel

- Dieb verliert wenn er am Rundenende auf Knoten mit Polizisten steht

Evaluation der Übung