Algorithmen für schwierige Probleme

Britta Dorn

Wintersemester 2011/12

24. November 2011

Farbkodierung — Beispiel Longest Path

Longest Path

gegeben: $G = (V, E)$ und $k \in \mathbb{N}$.
Frage: Gibt es einen **einfachen** Pfad der Länge k in G, d.h., Pfad aus k Knoten, so dass kein Knoten mehrfach im Pfad vorkommt?

Bemerkung: kann $k$te Potenz der Adjazenzmatrix bilden:
Eintrag $\neq 0$: Knoten sind durch Pfad der Länge k verbunden.
Problem: Pfad ist nicht unbedingt einfach.

Deshalb jetzt mit Farbkodierung!

Farbkodierung — Beispiel LONGEST PATH

Grundidee (randomisierter Ansatz)

Färbe Knoten von G per Zufall mit k verschiedenen Farben. Hoffe, dass alle Knoten eines Pfades unterschiedliche Farben bekommen.

Wir nennen einen Pfad *vollbunt*, wenn jeder seiner Knoten mit anderen Farbe gefärbt wurde.

Klar: Pfad vollbunt $\Rightarrow$ Pfad ist einfach.

Was lässt sich für einfache Pfade sagen? Ist ein einfacher Pfad immer vollbunt, wenn zufällig gefärbt wird?
Nein. Aber er ist vollbunt mit Wahrscheinlichkeit $\frac{k!}{k^k}$
(k^k: # Möglichkeiten, k Knoten mit k Farben zu färben.
$k!$: Die Reihenfolge spielt keine Rolle, $123 = 132$ usw.)
Es gilt: $\frac{k!}{k^k} > \frac{1}{e^k}$ (Stirling-Formel)

Farbkodierung — Beispiel LONGEST PATH

Lemma

Sei $G = (V, E)$ ein Graph, V sei mit k Farben gefärbt. Dann kann ein vollbunter Pfad aus k Knoten (falls er existiert) in Zeit $O(2^k \cdot |E|)$ gefunden werden.

Beweis: Algorithmus, der dynamisches Programmieren benutzt. Der Algorithmus findet alle vollbunten Pfade aus k Knoten, die bei einem festen Knoten s beginnen.

Farbkodierung — Beispiel Longest Path

Erster Algorithmus — ohne colour coding
Wir speichern in jedem Knoten v des Graphen die Menge $M(v)$ aller Pfade, die in diesem Knoten enden.

Algorithmus

- Starte mit Knoten s
- Annahme:
  - Pfad P der Länge i schon gefunden
  - P endet in v, also $P \in M(v)$.
- Suche nun Pfad P' der Länge $i + 1$. Der Knoten v war bisher das Ende des Pfades.
  - Betrachte Nachbar u von v.
  - Wenn $u \notin P$, dann
    - $P' := P \cup \{u\}$
    - $M(u) := M(u) \cup P'$

Beispiel Tafel.

Farbkodierung — Beispiel LONGEST PATH

Erster Algorithmus — ohne colour coding
Wie viele Pfade der Länge i kann es geben?
$\binom{n}{i}$ (# aller i-elementigen Teilmengen einer n-elementigen Menge)

Laufzeit

$O\left(\sum_{i=1}^{k} i \cdot \binom{n}{i} \cdot m\right)$
(i: prüfen, ob Knoten bereits in Knotenmenge vorkommt
$\binom{n}{i}$
$m = |E|$: alle Nachbarn des Knotens anschauen)

$= O(k \cdot m \cdot 2^n)$

Farbkodierung — Beispiel Longest Path

Zweiter Algorithmus — mit colour coding

Wir speichern in jedem Knoten v des Graphen die Menge $M_f(v)$ aller Pfade , die in diesem Knoten enden. Jetzt aber: nicht die Knoten des Pfades, sondern nur die Farben, die in dem Pfad vorkommen. (Achtung, Pfad muss nicht eindeutig sein.)

Algorithmus

- Starte mit Knoten s
- Annahme:
  - Vollbunter Pfad F der Länge i schon gefunden
  - F endet in v, also F (Farben des Pfades) $\in M_f(v)$.
- Suche nun Pfad F' der Länge $i+1$. Der Knoten v war bisher das Ende des Pfades.
  - Betrachte Nachbar u von v.
  - Wenn $c(u) \notin F$, dann
    - $F' := F \cup \{u\}$
    - $M_f(u) := M_f(u) \cup F'$

Farbkodierung — Beispiel Longest Path

Zweiter Algorithmus — mit colour coding
Wie viele vollbunte Pfade der Länge i kann es geben?
$\binom{k}{i}$ (# aller i-elementigen Teilmengen einer k-elementigen Menge)

Laufzeit

$O\left(\sum_{i=1}^{k} i \cdot \binom{k}{i} \cdot m\right)$
(i: prüfen, ob Knoten bereits in Knotenmenge vorkommt
$\binom{k}{i}$
$m = |E|$: alle Nachbarn des Knotens anschauen)

$= O(k \cdot m \cdot 2^k)$

Also: Statt Menge aller Knoten (bis zu n viele) zu betrachten, schauen wir nur Menge aller Farben (bis zu k viele) an, deshalb die bessere Laufzeit.

Farbkodierung — Beispiel Longest Path

Theorem

Longest Path kann in “erwarteter” Laufzeit $2^{O(k)} \cdot m$ gelöst werden.

Beweis: Will einfachen Pfad aus k Knoten. Färbe Graph zufällig mit k Farben. Falls es einen einfachen Pfad im Graphen gibt, wird er mit Wahrscheinlichkeit $\frac{k!}{k^k} > \frac{1}{e^k}$ vollbunt gefärbt, und mit dem Lemma kann man ihn dann auch finden. (Lemma: wenn es vollbunten Pfad gibt, finden wir ihn in $2^{O(k)} \cdot m$.).

Also:

- Färbe.
- Prüfe mit Lemma: vollbunt?
  - nein: dann war die Färbung nicht gut, färbe noch mal neu.
  - ja: vollbunter Pfad gefunden, also ist er auch einfach, fertig.

Wenn man $> e^k$ mal färbt, sollte der einfache Pfad vollbunt werden. (Erwartungswert ist > 1).

$e^k = 2^{O(k)}$ ($\rightarrow$ Tafel)

Kann also Longest Path in erwarteter Laufzeit $e^k O(2^k \cdot m) = 2^{O(k)} \cdot m$ lösen.

Farbkodierung — Beispiel Longest Path

Bemerkung: Dieser Algorithmus ist randomisiert
("Färbe zufällig..." "erwartete Laufzeit"...)

Kann ihn aber mittels Hashing (genauer: k-perfekte Familien von Hashfunktionen) derandomisieren
(bekomme dann deterministischen Algorithmus,
Laufzeiteinbußen...) ($\rightarrow$ Tafel)
kann damit zeigen:

Theorem

Longest Path kann deterministisch in $2^{O(k)} \cdot m \cdot \log n$ gelöst werden. Insbesondere ist Longest Path FPT bezüglich Parameter k (Länge des Pfades).

Baumzerlegungen

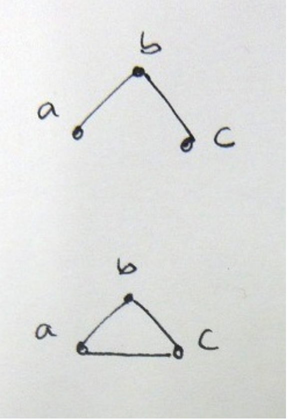
Tree decompositions of graphs

Definition (tree decomposition, treewidth)

Properties 1–3, explained in an easier way:

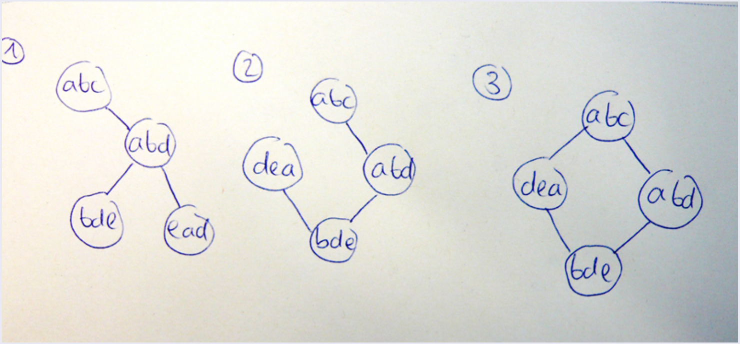
1. Each vertex of G appears in at least one of the bags
2. The endpoints of each edge appear together in at least one of the bags.
3. For each graph vertex v:
   The bags containing v form a subtree of T.

Übung: Baumzerlegungen



Quiz

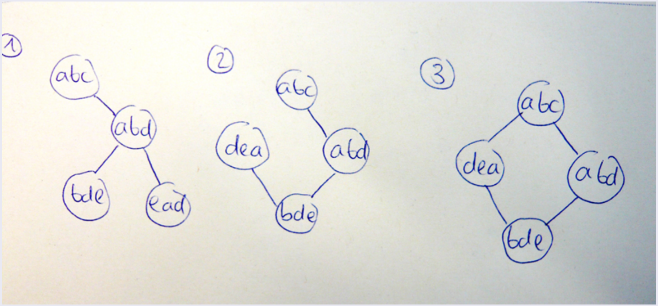
Welche dieser Zerlegungen sind gültige Baumzerlegungen?



Lösung

Quiz

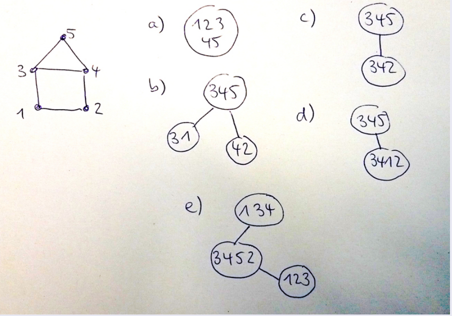
Welche dieser Zerlegungen sind gültige Baumzerlegungen?



keine

Quiz

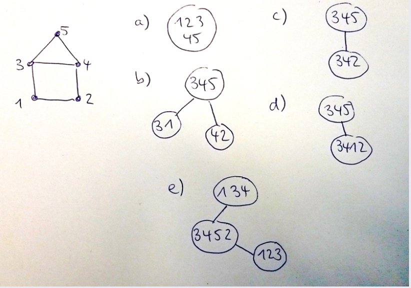
Welche dieser Zerlegungen sind gültige Baumzerlegungen?



Lösung

Quiz

Welche dieser Zerlegungen sind gültige Baumzerlegungen?



a), d)
Optimale Zerlegung?

Baumzerlegungen

8.5 Bemerkungen

- Baum hat Baumweite 1
- Eine Clique aus n Knoten hat Baumweite $n - 1$
- Jeder Graph mit n Knoten hat eine triviale Baumzerlegung der Weite $n - 1$
- Je kleiner die Baumweite eines Graphen, desto baumähnlicher ist der Graph
- I.A. ist es NP-schwer, eine optimale Baumzerlegung eines Graphen zu berechnen. (Aber: FPT bzgl. der Baumweite!) Offene Frage: Kann die Baumweite für planare Graphen in Polynomzeit berechnet werden?

8.6 Robber-cop game

Ein Räuber: bewegt sich von Knoten zu Knoten entlang der Kanten des Graphs, kann beliebig schnell laufen.

Mehrere Polizisten: bewegen sich von Knoten zu Knoten mit Hubschraubern (langsam), und sie sehen jedezeit, wo der Räuber sich befindet.

Die Baumweite eines Graphen ist die minimale Anzahl Polizisten, die man braucht, um den Räuber zu fangen, minus 1.

Quiz

Berechne die Baumweite dieser Graphen: