Karlsruher Institut für Technologie
Fakultät für Informatik
ITI Wagner

Algorithmen für Planare Graphen
Sommersemester 2022
Lars Gottesbüren, Laura Merker, Torsten Ueckerdt

Siebtes Übungsblatt

Ausgabe:	20. Juli 2022
Besprechung:	28. Juli 2022

1 Kreise

Sei G ein planarer Graph mit fester Einbettung und F die Menge der Facetten, wobei $f_0 \in F$ die äußere Facette bezeichnet. Bezeichne weiter $\operatorname{dist}(f)$ die Entfernung einer Facette f zur äußeren Facette, das hieß die Länge eines kürzesten Weges von f zu f_0 im Dualgraph. Sei $d = \max_{f \in F} \operatorname{dist}(f)$. Für $i = 1, \ldots, d$ sei E_i die Menge aller Kanten, bei denen beide inzidenten Facetten Entfernung i zur äußeren Facette haben. Außerdem enthalte für $i < d$ die Menge $E_{i,i+1}$ alle Kanten, bei denen eine angrenzende Facette Entfernung i zur äußeren Facette hat, die andere $i + 1$.

(i) Zeigen Sie, dass die Mengen $E_1, \ldots, E_d, E_{1,2}, \ldots, E_{d-1,d}$ die Kantenmenge von G partitionieren.

(ii) Zeigen Sie, dass jedes E_i eine kanten-disjunkte Vereinigung von einfachen Kreisen ist. Konstruieren Sie einen (innen triangulierten) planaren Graphen mit Einbettung, bei dem die Vereinigung nicht knoten-disjunkt ist.

(iii) Zeigen Sie, dass $E_1, \ldots, E_d$ paarweise disjunkt sind.

(iv) Geben Sie einen Algorithmus mit linearer Laufzeit an, der zu einem gegebenen Graphen G mit fester Einbettung die Kantenmengen $E_1, \ldots, E_d$ und $E_{1,2}, \ldots E_{d-1,d}$ bestimmt.

2 Spezialfall von s-t-Wegen

Geben Sie drei Algorithmen an, die folgendes Problem in Linearzeit lösen, darunter ein einfacher, der keinen der großen Algorithmen aus der Vorlesung als Baustein nutzt.

Gegeben ein planarer Graph G mit fester Einbettung und ausgezeichneten Knoten s und t, die an der äußeren Facette liegen, bestimme eine maximale Anzahl von paarweise kanten-disjunkten s-t-Wegen in G.

3 Right-First Tiefensuche und Left-First Breitensuche

Sei G ein ungerichteter, zusammenhängender, planar eingebetteter Graph und G^* der zugehörige Dualgraph. Sei u ein Knoten von G und e eine zu u inzidente Kante. Wir betrachten Graphsuchen, die bei u mit Kante e beginnen.

Algorithmus 1: Right-First-DFS

Füge neuen Knoten s und orientierte Kante (s, u) im Gegenuhrzeigersinn vor e an u ein
Lege (s, u) auf einen Stack
Solange *Stack nicht leer* **wiederhole**
- Betrachte oberste Kante (x, y)
- **Wenn** y *inzident zu nichtorientierter Kante*
  - Orientiere im Gegenuhrzeigersinn bzgl. y nächste nichtorientierte Kante $y \to w$ und lege diese auf den Stack
- **sonst**
  - Entferne (x, y) vom Stack

Algorithmus 2: Left-First-BFS

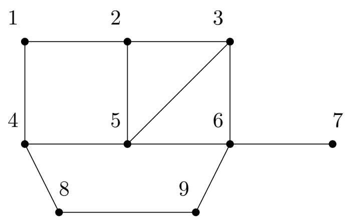
Orientiere alle zu u inzidenten Kanten $u \to w$ und füge diese, beginnend bei e, im Uhrzeigersinn bzgl. u in eine Queue ein
Solange *Queue nicht leer* **wiederhole**
- Betrachte erste Kante (x, y)
- **Für** *alle nichtorientierten Kanten* yw *im Uhrzeigersinn bzgl.* y
  - Orientiere $y \to w$ und füge (y, w) in die Queue ein
- Entferne (x, y) aus der Queue

Bei diesen beiden Suchen wird jeweils eine Reihenfolge $R = (e_1, \ldots, e_m)$ der Kanten festgelegt. Wir betrachten nun eine Right-First-Tiefensuche in G, beginnend bei Knoten u und Kante $e = \{u, v\}$ und erhalten eine Kantenfolge $R = (e = e_1, \ldots, e_m)$. Sei e^* die Dualkante zu e und f die Facette rechts von (u, v). Wir führen eine Left-First-Breitensuche auf G^* aus, beginnend mit f und e^*, und erhalten eine Kantenfolge $R^* = (e^* = e_1^*, \ldots, e_m^*)$. Die Reihenfolgen R und R^* heißen *dual*, falls e_i Dualkante zu e_i^* für alle $1 \leq i \leq m$.

(i) Wofür brauchen wir die Kante (s, u) in der Tiefensuche?

(ii) Bestimmen Sie für den zweidimensionalen Würfel Q_2 sowie untenstehenden Graphen die Reihenfolgen R und R^*, beginnend mit Knoten 1 und Kante $\{1, 2\}$.

Hinweis: Die Reihenfolgen sind dual.



(iii) Geben Sie einen Graphen an, für den R und R^* nicht dual sind.

Hinweis: Betrachten Sie Graphen mit Brücken und Kreisen.