

Sechstes Übungsblatt

Ausgabe: 9. Juli 2019
Besprechung: 11. Juli 2019

1 Right-First Tiefensuche und Left-First Breitensuche

Sei G ein ungerichteter, zusammenhängender, planar eingebetteter Graph, G^* der zugehörige Dualgraph und $e = (u, v)$ eine orientierte Kante.

<hr/> Algorithmus 1: Right-First-Kanten-DFS <hr/>	<hr/> Algorithmus 2: Left-First-Kanten-BFS <hr/>
Füge Kante (t, u) , $t \notin V$ im Gegenuhrzeigersinn vor e an u ein Betrachte e als nicht orientiert Lege (t, u) auf einen Stapel Solange <i>Stapel nicht leer</i> wiederhole Betrachte oberste Kante (x, y) Falls y <i>inzident zu nichtorientierter Kante</i> Orientiere im Gegenuhrzeigersinn bzgl. y nächste nichtorientierte Kante $y \rightarrow w$ und lege diese auf den Stapel sonst Entferne (x, y) vom Stapel	Orientiere alle zu u inzidenten Kanten $u \rightarrow w$ und hänge diese, beginnend bei e , im Uhrzeigersinn bzgl. u an eine Warteschlange Solange <i>Warteschlange nicht leer</i> wiederhole Betrachte erste Kante (x, y) Falls y <i>inzident zu nichtorientierter Kante</i> Orientiere alle solche Kanten $y \rightarrow w$ und hänge diese im Uhrzeigersinn bzgl. y an die Warteschlange sonst Entferne (x, y) aus der Warteschlange

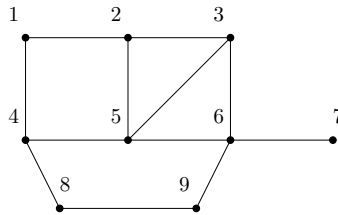
Hinweis: Die hinzugefügte Kante (t, u) von einem neuen Knoten t sorgt dafür dass die Right-First-Kanten-DFS den gesamten Graphen traversiert, falls e eine Brücke ist, die Suche also bei u fortgesetzt wird. Insbesondere wird e als erste Kante in der gegebenen Orientierung (u, v) gefunden.

In diesen beiden Kantensuchen wird jeweils eine Reihenfolge $R = (e_1, \dots, e_m)$ der Kanten festgelegt. Die zu e „rechte“ Facette sei f_1 , und die Dualkante $e^* = (f_1, f_2)$ zu e sei orientiert ausgehend von f_1 .

Wir betrachten eine Right-First-Tiefensuche in G , beginnend bei Kante e , mit Kantenfolge $R = (e = e_1, \dots, e_m)$ und eine Left-First-Breitensuche in G^* , beginnend bei Kante e^* , mit Kantenfolge $R^* = (e^* = e_1^*, \dots, e_m^*)$. Die Reihenfolgen R und R^* heißen dual, falls e_i Dualkante zu e_i^* für alle $1 \leq i \leq m$.

- (a) Bestimmen Sie für den zweidimensionalen Würfel Q_2 sowie untenstehenden Graphen die Reihenfolgen R und R^* bei beliebiger Startkante e .

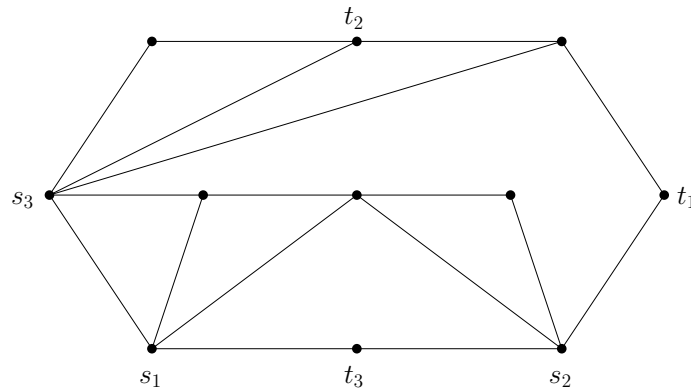
Hinweis: Die Reihenfolgen sind dual.



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

Hinweis: Betrachten Sie Graphen mit Brücken und Kreisen.

2 Kantendisjunkte Wegpackung



Betrachten Sie obenstehenden Graphen mit der angegebenen Menge an Terminalpaaren, und überprüfen Sie die *Geradheitsbedingung*. Ist das induzierte *kantendisjunkte* Wegpackungsproblem lösbar?

3 Knotendisjunkte Wegpackung

Definition (knotendisjunktes Wegpackungsproblem): Gegeben ein zusammenhängender, planarer Graph G mit fester Einbettung und paarweise verschiedenen Terminalen $s_1, t_1, s_2, t_2, \dots, s_k, t_k$ an der äußeren Facette. Gesucht sind paarweise knotendisjunkte s_i-t_i -Wege in G ($1 \leq i \leq k$).

Das Problem heißt *topologisch lösbar*, falls die Reihenfolge der Terminale „um die äußere Facette herum“ die Lösbarkeit nicht ausschließt.

- (a) Geben Sie für jede natürliche Zahl $k \geq 2$ einen zusammenhängenden, planaren Graphen G mit k Terminalpaaren (s_i, t_i) an der äußeren Facette an, sodass das knotendisjunkte Wegpackungsproblem

- nicht topologisch lösbar ist
- zwar topologisch lösbar ist, aber dennoch keine paarweise knotendisjunkten s_i-t_i -Wege in G existieren.

Bitte wenden

- (b) Sei o die Anzahl der Knoten von G , die zur äußeren Facette der Einbettung inzident sind (es gilt also $o \geq 2k$). Geben Sie einen Algorithmus an, der in Zeit $\mathcal{O}(o)$ feststellt, ob das knotendisjunkte Wegpackungsproblem topologisch lösbar ist.
- (c) Angenommen, das Problem sei topologisch lösbar. Geben Sie einen Linearzeitalgorithmus an, der entweder k knotendisjunkte s_i - t_i -Wege findet oder feststellt, dass das Problem nicht lösbar ist.
- (d) Zeigen Sie: Für einen planaren Graphen G mit Terminalpaaren an der äußeren Facette ist das knotendisjunkte Wegpackungsproblem genau dann *nicht lösbar*, wenn es nicht topologisch lösbar ist oder eine einfache geschlossene Kurve K in der Ebene existiert, die G nur in Knoten berührt, sodass die Anzahl der Terminalpaare mit genau einem Terminal innerhalb K größer als die Anzahl der ‚Berührknoten‘ ist.