Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

Institut für Theoretische Informatik
Lehrstuhl Prof. Dr. D. Wagner

Übungsblatt 2

Vorlesung Algorithmentechnik im WS 08/09

Ausgabe 04. November 2008
Abgabe 18. November, 15:30 Uhr (im Kasten vor Zimmer 319, Informatik-Hauptgebäude, 3. OG)

Bitte schreiben Sie nur Ihren Namen und keine Matrikelnummer auf Ihr Übungsblatt. Die Bearbeitung in Zweiergruppen ist ausdrücklich erwünscht.

Problem 1: Boruvka-MST

Sei $G = (V, E)$ ein einfacher, ungerichteter, zusammenhängender Graph mit **injektiver** Kostenfunktion $w : E \to \mathbb{N}^+$. Wir schreiben (u, v) für eine ungerichtete Kante $\{u, v\}$. Wir bezeichnen eine Kante (u, v) als *lokal minimal* wenn sie für u oder v unter allen Kanten, die mit diesem Knoten inzidieren, das kleinste Gewicht hat.

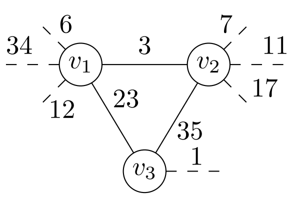
(a) Zeigen sie: Jede lokal minimale Kante gehört zu allen MST von G.

(b) Die Menge aller lokal minimalen Kanten eines Graphen ist ein Wald.

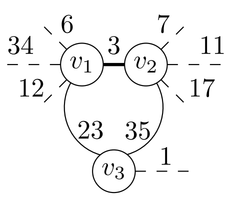
Eine *Boruvka-Phase* ist ein Algorithmus, der Folgendes leistet:

Schritt (1) Er bestimmt die Menge $L(G)$ aller lokal minimalen Kanten von G.

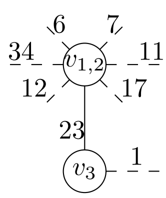
Schritt (2) Er berechnet den kontrahierten Graphen $G/L(G)$, wie in der Abbildung unten illustriert. Falls Mehrfachkanten entstünden, wird nur die Kante mit geringsten Gewicht behalten, und die anderen entfernt.



Der Graph vor der Boruvka Phase. Kante mit Gewicht 3 ist lokal minimal.



Knoten v_1 und v_2 werden entlang Kante mit Gewicht 3 kontrahiert.



Knoten v_1 und v_2 kontrahiert; teurere, doppelte Kante zu v_3 gelöscht.

Der so aus G entstehende Graph werde mit $B(G)$ bezeichnet.

(c) Implementieren sie in Pseudocode die Boruvka-Phase mit einer Worst-case-Laufzeit von $O(m)$ Vergleichsoperationen. Begründen Sie die Laufzeit. Ihnen steht eine Funktion LöscheTeureMehrfachkanten(G) zur Verfügung, die (im Worst-case) mit $O(m)$ Vergleichsoperationen auskommt.

(d) Zeigen sie, dass durch eine Boruvka-Phase die Anzahl der Knoten um mindestens die Hälfte reduziert wird.

Man kann zeigen, dass für jeden Graphen G gilt: Die Kanten in $L(G)$ und die Kanten in einem MST von $B(G)$ bilden zusammen einen MST von G.

(e) Nutzen sie Teilaufgabe (c) um einen MST mit $O(m \log n)$ Vergleichsoperationen im Worst-case zu bestimmen (Pseudocode). Begründen sie die Worst-case-Laufzeit von $O(m \log n)$.

Problem 2: Schnitte und MSTs

(a) Zeigen Sie: Wenn in einem Graph G mit reellen Kantengewichten für jeden Schnitt die den Schnitt kreuzende Kante minimalen Gewichts eindeutig ist, dann hat G einen eindeutigen aufspannenden Baum minimalen Gewichts.

(b) Gilt die Umkehrung von (a)?

(c) Zeigen oder widerlegen Sie: Wenn in einem Graph G mit reellen Kantengewichten alle Kanten paarweise verschiedene Gewichte haben, dann hat G einen eindeutigen aufspannenden Baum minimalen Gewichts.

(d) Zeigen oder widerlegen Sie die Umkehrung der Aussage in (c).

Problem 3: Vier Äquivalente Aussagen

Zur Erinnerung: Ein Zyklus ist eine Knotenfolge, bei der Anfangs- und Endknoten übereinstimmen und zwischen je zwei aufeinanderfolgenden Knoten eine Kante existiert. Ein Kreis ist ein Zyklus, bei dem keine Knoten außer dem Anfangsknoten mehrmals vorkommen. In jedem Zyklus ist ein Kreis enthalten.

Zeigen Sie, dass die folgenden Aussagen für einen ungerichteten Graphen $G = (V, E)$ äquivalent sind.

(a) G ist zusammenhängend und enthält keine Kreise.

(b) G ist maximal kreisfrei, d.h. G enthält keinen Kreis und für je zwei Knoten $u, v \in V$ mit $\{u, v\} \notin E$ enthält der Graph $G' = (V, E \cup \{\{u, v\}\})$ einen Kreis.

(c) Zwischen je zwei Knoten in V gibt es genau einen Weg in G.

(d) G ist minimal zusammenhängend, d.h. G ist zusammenhängend, und für jede Kante $e \in E$ ist der Graph $G' = (V, E \setminus \{e\})$ nicht zusammenhängend.

Problem 4: Einmaschinenscheduling

Eine Reihe von Aufträgen $A_1, \ldots A_n$ müssen von einer einzigen Maschine abgearbeitet werden, es kann also zu jedem Zeitpunkt höchstens ein Auftrag bearbeitet werden. Alle Aufträge benötigen die gleiche Bearbeitungszeit. Jeder Auftrag A_i hat eine Deadline $D_i \in \mathbb{R}$ bis zu der er fertig sein muss.

(a) Zeigen Sie, dass die Menge aller Teilmengen von Aufträgen, die rechtzeitig bearbeitet werden können, ein Matroid ist.

(b) Wird ein Auftrag nicht rechtzeitig fertig, so muss eine Strafe P_i bezahlt werden, deren Höhe vom jeweiligen Auftrag abhängt. In welcher Reihenfolge sollten die Aufträge abgearbeitet werden, damit die Gesamtstrafe minimiert wird?

Hinweis/Spoiler: Sei o.B.d.A. $D_1 \leq D_2 \leq \cdots \leq D_n$. Eine Teilmenge $\{A_{i_1}, \ldots, A_{i_k}\}$ von k Aufträgen ($i_1 \neq \cdots \neq i_k \in \{1, \ldots, n\}$) kann rechtzeitig bearbeitet werden, wenn für alle $j \leq k$ gilt : $D_{i_j} \geq j$.