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

Übungsblatt 3

Vorlesung Algorithmentechnik im WS 09/10

Ausgabe 19. November 2009
Abgabe 03. Dezember, 15:30 Uhr in der Übung (oder im Kasten vor Zimmer 319, Geb. 50.34)

Bitte schreiben Sie Ihren Namen und Ihre E-Mailadresse aber keine Matrikelnummer auf Ihr Übungsblatt. Die Bearbeitung in Zweiergruppen ist ausdrücklich erwünscht.

Problem 1: Kreuzende Schnitte – Schnitte in Graphen [vgl. Kapitel 3 im Skript]

Zwei Schnitte $(S, V \setminus S)$ und $(T, V \setminus T)$ in einem (ungerichteten) Graphen $G = (V, E)$ *kreuzen sich*, wenn keine der Mengen $A := S \cap T$, $B := S \setminus T$, $C := T \setminus S$ und $D := V \setminus (S \cup T)$ leer ist. Sei $c : E \to \mathbb{R}_0^+$ eine Kantengewichtsfunktion auf G.

(a) Seien $(S, V \setminus S)$ und $(T, V \setminus T)$ zwei sich kreuzende s-t-Schnitte mit $s \in S$ und $t \in T$. Zeigen Sie: Es gilt $s \in B$ und $t \in C$.

(b) Seien $(S, V \setminus S)$ und $(T, V \setminus T)$ zwei sich kreuzende minimale s-t-Schnitte mit $s \in S$ und $t \in T$. Zeigen Sie: $(B, V \setminus B)$ und $(C, V \setminus C)$ sind zwei kreuzungsfreie minimale s-t-Schnitte.

Problem 2: Algorithmus von Goldberg und Tarjan – Maximale Flüsse [vgl. Kapitel 4.2.3 im Skript]

Die Grundidee des Algorithmus von Goldberg und Tarjan basiert auf Präflüssen und Push- und Relabel-Operationen. Algorithmen, die dieses Grundkonzept nutzen, nennt man daher auch *Preflow-Push*- oder *Push-Relabel*-Algorithmen.

Gegeben sei nun ein Flussnetzwerk $(D; s, t; c)$ bzw. dessen Erweiterung $(D'; s.t; c')$ und ein darin mit einem *Push-Relabel*-Algorithmus berechneter maximaler Fluss (d.h. die Abbildungen f, r_f, e und *dist* stehen bezüglich des fertig berechneten Flusses zur Verfügung).

Entwickeln Sie einen möglichst schnellen Algorithmus um die Knotenpartition eines zugehörigen minimalen s-t-Schnitts (S, T) zu berechnen. Geben Sie Ihren Algorithmus in Pseudocode an. Begünden Sie die Laufzeit und die Korrektheit Ihres Algorithmus (Hinweis: Es gibt einen Algorithmus in $O(|V|)$).

Problem 3: Gomory-Hu-Bäume – Schnitte in Graphen [vgl. Kapitel 3 im Skript]

Gegeben sei ein ungerichteter, nicht-negativ gewichteter, zusammenhängender Graph $G = (V, E)$. Ein zu G gehöriger Gomory-Hu-Baum ist ein ungerichteter, nicht-negativ gewichteter, zusammenhängender Baum $T(G)$ über der Knotenmenge V so, dass gilt:

(a) jede Kante $\{u, v\}$ in $T(G)$ induziert einen minimalen u-v-Schnitt in G, indem $T(G)$ beim Entfernen von $\{u, v\}$ in zwei Teilbäume zerfällt, deren Knotenmengen gerade die beiden Seiten des Schnitts darstellen.

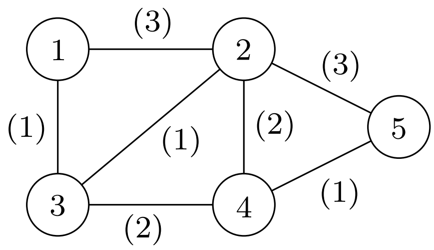
(b) das Gewicht einer Kante $\{u, v\}$ in $T(G)$ entspricht dem Gewicht des durch $\{u, v\}$ induzierten minimalen u-v-Schnitts in G.

Bemerkung: Die Kantemenge eines Gomory-Hu-Baumes $T(G)$ muss **keine** Teilmenge der Kantenmenge des zugrundeliegenden Graphen G sein. Zu jedem ungerichteten, nicht-negativ gewichteten, zusammenhängenden Graphen existiert mindestens ein Gomory-Hu-Baum.

Zeigen Sie: Für zwei beliebige Knoten $s, t \in V$ induziert eine auf dem s-t-Pfad in $T(G)$ minimal gewichtete Kante einen minimalen s-t-Schnitt in G.

Problem 4: Algorithmus von Stoer & Wagner – Schnitte in Graphen [vgl. Kapitel 3.2 im Skript]

(a) Wenden Sie auf den unten abgebildeten Graphen (Kantengewichte in Klammern) den Algorithmus von Stoer & Wagner an. Geben Sie nach jeder Phase die Knoten s und t, den Schnitt der Phase und dessen Gewicht an und zeichnen Sie den nach dem Verschmelzen resultierenden Graphen (mit Kantengewichten). **Verwenden Sie in Phase i den Knoten als Startknoten, der Knoten i des Originalgraphen enthält.** Geben Sie zum Schluss den minimalen Schnitt S_{min} an.



(b) Liefert der Algorithmus von Stoer & Wagner auch für negative Kantengewichte einen global minimalen, nichttrivialen Schnitt? Begründen Sie Ihre Antwort.