Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

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

Übungsblatt 3

Vorlesung Algorithmentechnik im WS 08/09

Ausgabe 18. November 2008
Abgabe 2. Dezember, 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: Kreuzende Schnitte

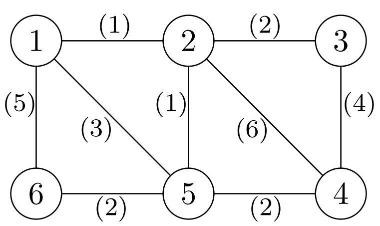
Zwei Schnitte $(S, V \setminus S)$ und $(T, V \setminus T)$ in einem Graph $G = (V, E)$ *kreuzen sich*, wenn keine der Mengen $A := S \cap T$, $B := S \setminus T$, $C := T \setminus S$ und $D := V \setminus (S \cup T)$ leer ist. Sei $c : E \to \mathbb{R}_0^+$ eine Kantengewichtsfunktion auf G. Zeigen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

(a) Sind $(S, V \setminus S)$ und $(T, V \setminus T)$ zwei sich kreuzende Schnitte minimalen Gewichts λ in G, so gilt $c(A, D) = c(B, C) = 0$ und $c(A, B) = c(B, D) = c(D, C) = c(C, A) = \lambda/2$.

(b) Sind s und t zwei adjazente Knoten mit $c(\{s, t\}) > 0$, so enthält die Menge der minimalen Schnitte von G, die s und t trennen, keine zwei sich kreuzenden Schnitte.

Problem 2: Stoer-Wagner

(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.



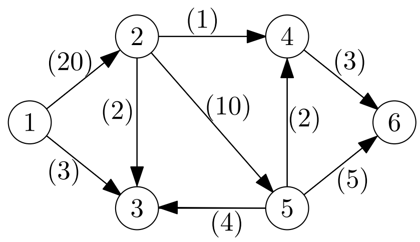
(b) An welcher Stelle im Korrektheitsbeweis zum Algorithmus von Stoer und Wagner wurde verwendet, dass die Kantengewichte nicht negativ sind? Finden Sie ein Beispiel eines Graphen mit negativen Kantengewichten und einen Startknoten a, so dass der Algorithmus von Stoer und Wagner keinen minimalen Schnitt liefert.

Problem 3: Flüsse

Gegeben sei ein Flussproblem in einem Netzwerk $D = (V, E)$, in dem es zu einigen Kanten (u, v) auch Kanten (v, u) gibt. Zeigen Sie, dass man dieses Flussproblem auf ein Flussproblem auf dem Netzwerk $D' = (V, E')$ überführen kann, wobei gilt: E' ist maximale Teilmenge von E mit $(u, v) \in E' \Rightarrow (v, u) \notin E'$, so dass der Wert des Maximalflusses nicht verändert wird.

Problem 4: Goldberg–Tarjan

Bestimmen Sie den maximalen Fluss in dem unten angegebenen Netzwerk mit $s = 1$ und $t = 6$ (Kantenkapazitäten in Klammern). Benutzen Sie hierzu den Algorithmus von Goldberg–Tarjan. Geben Sie **sämtliche** Zwischenschritte an. Zeichnen Sie das aktuelle Netzwerk **vor** jeder (bis auf die erste) RELABEL-Operation und nachdem der Algorithmus terminiert. Halten Sie sich an die Notation aus dem Skript.



Problem 5: Das Escape Problem

Gegeben seien $n \times n$ Gitterpunkte.

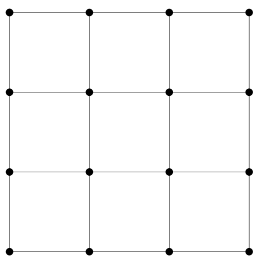


Abbildung 1: Gitternetzwerk für $n = 4$

Sei S eine Teilmenge der Gitterpunkte mit $|S| < n^2$. Das *Escape Problem* besteht darin zu entscheiden, ob es $|S|$ viele knotendisjunkte Wege gibt, so dass jeder Weg von einem Knoten aus S startet und an einem Randpunkt des Gitters endet.

(a) Geben Sie für $n = 4$ eine Ja-Instanz mit möglichst vielen Startknoten an, sowie eine Nein-Instanz mit möglichst wenig Startknoten.

Das maximale Flussproblem aus der Vorlesung kann folgendermaßen erweitert werden: Neben den Kantenkapazitäten c seien noch Knotenkapazitäten $\gamma : V \to \mathbb{R}_0^+$ gegeben. In einem solchen Netzwerk $(D; s; t; c; \gamma)$ heißt eine Abbildung f *Fluss*, wenn sie neben den bekannten Eigenschaften (Flusserhaltungsbedingung und (Kanten)-Kapazitätsbedingung) auch die folgende erfüllt:

- Für alle $i \in V$ ist die *Knotenkapazitätsbedingung*

$$\begin{aligned} \sum_{\{j|(i,j)\in E\}} f(i,j) \le \gamma(i) \quad &\text{wenn} \quad i \in V \setminus \{t\} \\ \sum_{\{j|(j,i)\in E\}} f(j,i) \le \gamma(i) \quad &\text{wenn} \quad i = t \end{aligned}$$

  erfüllt.

(b) Zeigen Sie, dass die Bestimmung eines maximalen Flusses in einem Netzwerk mit Kanten- **und** Knotenkapazitäten auf ein maximales Flussproblem in einem normalen (d.h. ohne Knotenkapazitäten) Netzwerk mit vergleichbarer Größe zurückgeführt werden kann.

(c) Wie würden Sie das Escape-Problem algorithmisch lösen? Geben Sie die Worst-case Laufzeit Ihres Ansatzes an. *Hinweis:* Pseudocode ist nicht gefordert!