Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Christian Scheideler
Stefan Eckhardt

Sommersemester 2006
Übungsblatt 2
10. Mai 2006

Effiziente Algorithmen und Datenstrukturen II

Abgabetermin: 17. Mai 2006 **vor** *der Zentralübung*

Aufgabe 1 (10 Punkte)

Ein zusammenhängender Graph $G = (V, E)$ heißt *k-fach-kantenzusammenhängend*, falls für jede Teilmenge $E' \subseteq E$ der Kardinalität $|E'| \leq k$ gilt: $G' = (V, E \setminus E')$ ist zusammenhängend. Geben Sie einen Algorithmus an, der in polynomieller Zeit zu einem zusammenhängenden Graphen $G = (V, E)$ und einer natürlichen Zahl k testet, ob G k-fach-kantenzusammenhängend ist. Beweisen Sie die Korrektheit Ihres Algorithmus.

Aufgabe 2 (10 Punkte)

In der Vorlesung wurde argumentiert, das Goldbergs (Preflow-Push-Methode) Maxflow-Algorithmus eine Lauzeit von $O(n^3)$ besitzt, falls die *Highest-Label-First* Heuristik zur Auswahl des aktiven Knotens verwendet wird. Beweisen Sie diese obere Schranke *formal*.

Aufgabe 3 (10 Punkte)

Ein Graph $G(V, E)$ heißt *bipartit*, wenn es eine Partition (V_1, V_2) von V mit $V_1 \cap V_2 = \emptyset$ und $V_1 \cup V_2$ gibt, so für alle Paare von Knoten $x, y \in V$ gilt: falls $x, y \in V_i$ für ein $i \in \{1, 2\}$ so folgt $(x, y) \notin E$. Ein *Maximum Weight Matching* in einem gewichteten Graphen $G = (V, E, w)$ ist eine Teilmenge $E' \subseteq E$ mit *maximalem Gewicht*, so dass für alle $e, f \in E$ gilt: $e \cap f = \emptyset$ (hierbei sei für alle $E' \subseteq E$ das Gewicht $w(E') = \sum_{e \in E'} w(e)$). Geben Sie einen Algorithmus an, der in polynomieller Zeit zu einem gegebenen bipartiten Graphen $G(V, E, W)$ ein *Maximum Weight Matching* berechnet.

Aufgabe 4 (10 Punkte)

Zum Abschluss des Kapitels „Flüsse“ nochmal ein Quiz, um Ihr Wissen zu überprüfen. Welche der folgenden Aussagen sind richtig, welche falsch? Beweisen Sie Ihre Antwort.

1. Ist f ein maximaler Fluß, so gilt entweder $f(e) = 0$ oder $f(e) = c(e)$ für jede Kante $e \in E$ eines Netzwerkes.

2. Jedes Netzwerk besitzt einen maximales Fluß f, für den entweder $f(e) = 0$ oder $f(e) = c(e)$ für jede Kante $e \in E$.

3. Wenn alle Kanten eines Netzwerkes paarweise verschiedene Kapazitäten besitzen, so gibt es einen eindeutig bestimmten minimalen Schnitt.

4. Wenn man in einem Netzwerk jede gerichtete Kante durch eine ungerichtete ersetzt, so bleibt der Wert des maximalen Flusses unverändert.

5. Wenn man jede Kantenkapazität mit einer positiven Zahl λ multipliziert, so bleibt der Wert des minimalen Schnittes unverändert.

6. Wenn man zu jeder Kantenkapazität eine positiven Zahl λ addiert, so bleibt der Wert des minimalen Schnittes unverändert.