



Algorithmische Mathematik I

Wintersemester 2011 / 2012

Prof. Dr. Sven Beuchler

Peter Zaspel



Übungsblatt 9.

Abgabe am **21.12.2011**.

Achtung: Die Abgabe der Programmieraufgaben beginnt in den CIP-Pools ab sofort stets am Mittwoch und endet am Dienstag der darauffolgenden Woche! Analog wird ab sofort auch die Anmeldeleiste ausgehängt. Somit wird die Liste am 21.12.2011 ausgehängt und am 10.01.2012 abgehängt. Die Abgabe der Aufgaben erfolgt dann vom 11.01. bis 17.01.2012.

Aufgabe 1. (Kapazitätsbeschränkte Knoten)

Es sei ein Fluss-Netzwerk $N = (G, c, s, t)$ gegeben. Wir wollen nun den Fall von kapazitätsbeschränkten Knoten betrachten. Eine sinnvolle Anwendung ist hier zum Beispiel ein Bewässerungs-Netzwerk, dessen Knoten Pumpstationen mit einer beschränkten Kapazität sind. Formal führen wir nun eine weitere Abbildung $d : V \rightarrow \mathbb{R}_0^+$ ein. Zusätzlich gelte nun für den Fluss f neben dem Kirchhoffschen Gesetz noch die Bedingung

$$\sum_{e \in \delta^+(v)} f(e) \leq d(v) \quad \text{für } v \neq s, t.$$

- a) Reduzieren Sie diese spezielle Variante eines Flussproblems auf die Ihnen bekannte Definition. Geben Sie also ein Verfahren / eine Vorschrift an, wie man dieses Flussproblem in ein normales Flussproblem überführen kann.
- b) Verallgemeinern Sie das Max-Flow-Min-Cut-Theorem bzgl. dieser neuen Flussdefinition und skizzieren Sie den dann notwendigen Beweis für das Theorem.

(10 Punkte)

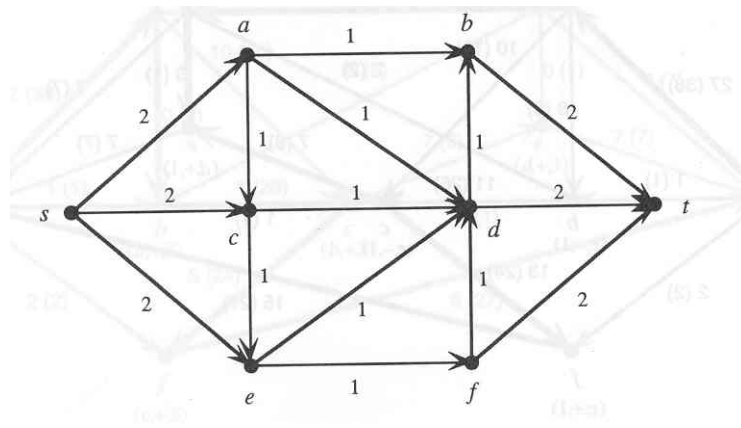
Aufgabe 2. (Edmonds-Karp)

Angenommen, man hat in einem Fluss-Netzwerk N einen maximalen Fluss mit dem Algorithmus von Edmonds und Karp bestimmt und stellt hinterher fest, dass die Kapazität einer Kante e irrtümlich falsch angesetzt wurde. Gibt es eine Möglichkeit, die einzelnen Resultate der vorangegangenen Ausführung von Edmonds-Karp erneut zu verwenden? Wenn dies der Fall ist, geben sie bitte einen groben Algorithmus an, mit dem ein entsprechendes Update möglich ist.

(5 Punkte)

Aufgabe 3. (Algorithmus von Edmonds und Karp)

Wenden Sie den Algorithmus von Edmonds und Karp auf das nachfolgend Netzwerk (bei Startknoten s und Zielknoten t) an.



(5 Punkte)

Programmieraufgabe 1. (Edmonds-Karp-Algorithmus)

In dieser Programmieraufgabe soll der Algorithmus von Edmonds und Karp zur Berechnung eines maximalen Flusses implementiert werden. Es können dabei die in der Vorlesung besprochenen Methoden, die in der Datei `NetworkFlow.cpp` (auf der Vorlesungshomepage) vorliegen, als Hilfestellung verwendet werden. Bitte beachten Sie, dass bereits der Methodenkopf für den Edmonds-Karp-Algorithmus definiert ist, Sie also nur noch die eigentliche Methode implementieren müssen. Außerdem müssen entsprechende Methoden zum Augmentieren der Kanten implementiert werden. Anstatt der im Hauptprogramm aufgerufenen Methoden `Netzwerk_gitter`, `Netzwerk_vollst`, etc. verwenden Sie bitte Ihre eigenen Graphengenerierungs-Routinen. Die verschiedenen Matrix- und Vektor-Routinen für CSR-Matrizen finden Sie ebenfalls auf der Vorlesungswebseite (unter `vector.h`). Die in `NetworkFlow.cpp` eingebundenen Headerdateien `kuerzwege.h` und `network.h` liegen nicht auf der Vorlesungswebseite vor, da hierin die noch von Ihnen zu implementierenden Methoden enthalten sind.

- a) Implementieren Sie den Edmonds-Karp-Algorithmus auf CSR-Adjazenz-Matrizen.
- b) Testen Sie den Algorithmus am Beispiel aus der Theorie-Aufgabe 3 und führen Sie wie immer den `valgrind`-Test durch.
- c) Lassen sie den Algorithmus auch auf dem 2D-Gitter-Graphen mit randomisierten Gewichten / Kapazitäten (bei dem Startknoten links oben und dem Zielknoten rechts unten) aus den vorangegangenen Übungszetteln laufen und messen und plotten Sie wie gewohnt die Laufzeit.

(15 Punkte)