Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Ernst W. Mayr
Matthias Baumgart

Sommersemester 2008
Übungsblatt 1
18. April 2008

Effiziente Algorithmen und Datenstrukturen II

Abgabetermin: 25.04.2008 **vor** *der Vorlesung*

Aufgabe 1 (10 Punkte)

Wir nennen einen Fluss f *gerade*, falls $f(a)$ für alle $a \in A$ gerade ist. Analog heißt ein Fluss f *ungerade*, falls $f(a)$ für alle $a \in A$ ungerade ist. Beweisen oder widerlegen Sie:

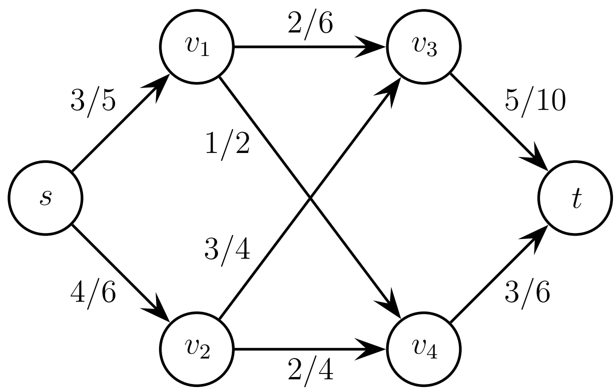
Sind alle Kapazitäten $c(a)$

(a) gerade, dann gibt es einen maximalen Fluss, der gerade ist.

(b) ungerade, dann gibt es einen maximalen Fluss, der ungerade ist.

Aufgabe 2 (10 Punkte)

Gegeben sei das folgende Netzwerk:



Die angegebenen Zahlen stellen die unteren bzw. oberen Kapazitätsschranken dar.

(a) Bestimmen Sie einen maximalen und einen minimalen Fluss von s nach t.

(b) Ändern Sie die Kapazität der Kante (v_3, t) von 5/10 zu 9/12. Gibt es nun noch einen zulässigen Fluss?

Aufgabe 3 (10 Punkte)

Sei $N = (V, A, c, s, t)$ ein Flussnetzwerk. Es seien $f_1, f_2 \colon A \to \mathbb{R}_0$ zwei Flüsse in N. Wir definieren Funktionen $f_1 + f_2$, $\alpha f_1 \colon A \to \mathbb{R}_0$ wie folgt:

$$\begin{aligned} (f_1 + f_2)(a) &=_{\text{def}} f_1(a) + f_2(a) \qquad \text{und} \\ (\alpha f_1)(a) &=_{\text{def}} \alpha \cdot f_1(a)\ . \end{aligned}$$

Zeigen Sie, dass die Flüsse in einem Netzwerk eine konvexe Menge bilden, d.h. für alle $0 \leq \alpha \leq 1$ und alle Flüsse f_1, f_2 ist $\alpha f_1 + (1 - \alpha) f_2$ wieder ein Fluss in N.

Aufgabe 4 (10 Punkte)

Ein *Matching* in einem ungerichteten Graphen $G = (V, E)$ ist eine Teilmenge $E' \subseteq E$, so dass je zwei verschiedene Kanten aus E' (knoten-)disjunkt sind. Ein ungerichteter Graph $G = (V, E)$ wird *bipartit* genannt, wenn es zwei Mengen V_1 und V_2 mit $V = V_1 \cup V_2$ und $V_1 \cap V_2 = \emptyset$ gibt, so dass für alle Kanten $\{u, v\} \in E$ entweder $u \in V_1$ und $v \in V_2$ oder $u \in V_2$ und $v \in V_1$ gilt.

Zeigen Sie, wie man mit Hilfe von Flussalgorithmen in einem gegebenen bipartiten Graphen $G = (V_1 \cup V_2, E)$ ein Matching E' maximaler Kardinalität (maximum Matching) finden kann.