Technische Universität München
Institut für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Angelika Steger

WS 1996/7
Übungsblatt 11
29. Januar 1996

# Effiziente Algorithmen und Datenstrukturen

Abgabe am 5. Februar 1996 bis 8:00 Uhr im Briefkasten

## Aufgabe 41

Geben Sie ein Beispiel eines Digraphen an, der keine negativen Kreise enthält, für den der Algorithmus von Dijkstra aber *keinen* kürzesten Pfad findet.

## Aufgabe 42

Gegeben sei ein azyklischer Digraph $D = (V, A)$ mit nichtnegativer Längenfunktion $\ell : A \to \mathbb{R}^+$ und zwei ausgezeichnete Knoten $s, t \in V$. Gesucht ist ein *längster* $s$-$t$-Pfad.
Überlegen Sie sich einen Algorithmus, der dieses Problem möglichst effizient löst. Fällt Ihnen auch ein Algorithmus für beliebige Digraphen ein?

## Aufgabe 43

Gegeben sei ein azyklischer Digraph $D = (V, A)$ mit nichtnegativer Längenfunktion $\ell : A \to \mathbb{R}^+$ und zwei ausgezeichnete Knoten $s, t \in V$. Gesucht ist ein zweit-kürzester $s$-$t$-Pfad im folgenden Sinne: Denken Sie sich alle $s$-$t$-Pfade nach ihrer Länge geordnet (wobei die Reihenfolge zwischen gleichlangen Pfaden egal ist); dann ist ein Pfad gesucht, der in einer solchen Ordnung an zweiter Stelle stehen kann (insbesondere ist ein solcher Pfad nicht länger als ein kürzester, wenn es mehrere kürzeste Pfade gibt).
Überlegen Sie sich einen Algorithmus, der dieses Problem in Laufzeit $\mathcal{O}(m + n \log n)$ löst. Können Sie Ihren Algorithmus auch so modifizieren, daß statt dem zweit-kleinsten der $k$'t-kleinste Pfad ausgegeben wird?

## Aufgabe 44

Gegeben sei ein Graph $G = (V, E)$ mit *nichtnegativer* Längenfunktion $\ell : E \to \mathbb{R}^+$ und eine (nichtleere) Knotenmenge $X \subseteq V$. Gesucht ist eine Partition $V = \bigcup_{x \in X} V_x$, so daß für alle $x \in X$ gilt:

$$v \in V_x \quad \Longrightarrow \quad \forall x' \in X : dist(v, x) \leq dist(v, x'),$$

wobei $dist(v, w)$ die Länge eines kürzesten $v$-$w$-Pfades (bzgl. der Längenfunktion $\ell$) bezeichne. (Jede der Mengen $V_x$ besteht also nur aus Knoten $v \in V$, deren Abstand zu $x$ höchstens so groß ist wie zu den übrigen Knoten aus $X$.)
Überlegen Sie sich einen Algorithmus, der dieses Problem in Laufzeit $\mathcal{O}(m + n \log n)$ löst.