Lehrstuhl für Informatik I
— Informationsstrukturen und
wissensbasierte Systeme —
Universität Würzburg

Würzburg, 11. Juni 1997

Prof. Dr. H. Noltemeier
Dipl. Math. S.O. Krumke

Übungen zur Vorlesung
Graphentheoretische Konzepte und Algorithmen (SS97)

5. Übungsblatt

Aufgabe 17: (Minimale Spannende Bäume)

Im folgenden sei $G = (V, R, \alpha, \omega)$ ein endlicher Graph und $c\colon R \to \mathbb{R}$ eine Gewichtsfunktion.

(a) Sei T ein minimaler spannender Baum von G und sei $L(T)$ die sortierte Liste seiner Pfeilgewichte, d.h. $L = (c(e_1), c(e_2), \dots, c(e_{n-1}))$, wenn $T = (V, E_T)$ mit $E_T = \{e_1, \dots, e_{n-1}\}$ und $c(e_1) \le c(e_2) \le \cdots \le c(e_{n-1})$ ist. Sei T' ein weiterer minimaler spannender Baum von G. Beweisen Sie, daß $L(T') = L(T)$ gilt. **(4 Punkte)**

(b) Sei $c\colon R \to \mathbb{R}$ injektiv. Beweisen Sie, daß dann der minimale spannende Baum bezüglich c eindeutig ist. **(1 Punkt)**

Aufgabe 18: (Flaschenhals-Bäume)

Sei im folgenden $G = (V, R, \alpha, \omega)$ ein endlicher schwach zusammenhängender Graph und $c\colon R \to \mathbb{R}$ eine Gewichtsfunktion. Für einen Partialgraphen $G' = G_{R'}$ von G definieren wir das *Flaschenhalsgewicht* (engl. *bottleneck weight*) $c_{\max}(G')$ als das Gewicht des schwersten Pfeils in G', d.h. $c_{\max}(G') := \max\{\, c(r) : r \in R' \,\}$. Ein minimaler spannender Flaschenhalsbaum (MBT) T^* von G ist ein spannender Baum mit $c_{\max}(T^*) \le c_{\max}(T)$ für jeden spannenden Baum T von G.

Zeigen Sie, daß jeder MST bezüglich der Gewichtsfunktion c auch ein MBT bezüglich c ist. **(4 Punkte)**

Aufgabe 19: (Schnelle MST- und Komponenten-Berechnungen)

Die große Ackermann-Funktion $\mathcal{A}$ ist für natürliche Zahlen $i, j \ge 1$ wie folgt definiert:

$$\begin{aligned} \mathcal{A}(1, j) &= 2^j && \text{falls } j \ge 1, \\ \mathcal{A}(i, 1) &= \mathcal{A}(i-1, 2) && \text{falls } i \ge 2, \\ \mathcal{A}(i, j) &= \mathcal{A}(i-1, \mathcal{A}(i, j-1)) && \text{falls } i, j \ge 2, \end{aligned}$$

Die *inverse Ackermann-Funktion* definiert man durch

$$\alpha(m, n) = \min\{\, i \ge 1 : \mathcal{A}(i, \lfloor m/n \rfloor) > \log_2 n \,\}.$$

Sei im folgenden wieder $G = (V, R, \alpha, \omega)$ ein endlicher schwach zusammenhängender Graph mit $n := |V|$ Ecken und $m := |R|$ Pfeilen.

(a) Zeigen Sie, daß man die schwachen Zusammenhangskomponenten von G in $\mathcal{O}(n + m\alpha(m, n))$ Zeit berechnen kann. **(2 Punkte)**

(b) Sei $c\colon R \to \{1, \ldots, n\}$ eine Pfeilgewichtungsfunktion. Geben Sie einen Algorithmus an, der einen MST von G bezüglich c in Zeit $\mathcal{O}(n+m\alpha(m, n))$ berechnet. **(1 Punkt)**

Bemerkung: Man sieht leicht, daß die Ackermann-Funktion A in jedem Argument monoton wachsend ist. Da

$$A(4,1) = A(3,2) = \left.2^{2^{\cdot^{\cdot^{2}}}}\right\}\text{ 16 mal}$$

weitaus größer als die geschätzte Anzahl der Atome im Universum (etwa 10^{80}) ist, gilt $\alpha(m, n) \leq 4$ für alle „in der Praxis vorkommenden Werte“ ($n \leq 10^{80}$).

Aufgabe 20: (Kürzeste Wege)

Sei im folgenden wieder $G = (V, R, \alpha, \omega)$ ein endlicher Graph und $\ell\colon R \to \mathbb{R}$ eine Gewichtsfunktion. Die *Länge* $\ell(w)$ eines Weges $w = (r_1, \ldots, r_k)$ in G ist dann $\ell(w) := \sum_{i=1}^{k} \ell(r_i)$. Der *Abstand* $\delta(u, v)$ zweier Knoten $u, v \in V$ ist dann die Länge eines kürzesten Weges von u nach v, sofern $v \in E_G(u)$ und $+\infty$ sonst.

Wir betrachten den folgenden Algorithmus zur Bestimmung der Abstände $\delta(s, \cdot)$ für eine gegebene Ecke s.

Input: Ein gerichteter Graph $G = (V, R, \alpha, \omega)$ in Adjazenzlistendarstellung; eine nichtnegative Gewichtsfunktion $\ell\colon R \to \mathbb{R}_{\geq 0}$ und ein Knoten $s \in V$

for all $v \in V$ **do**
  $d[v] \leftarrow +\infty$
  INSERT(Q, v) {Füge v mit Schlüssel $d[v]$ in die Prioritätsschlange Q ein}
  $\pi[v] \leftarrow$ NIL
$d[s] \leftarrow 0$
DECREASE-KEY$(Q, s, 0)$ {Vermindere den Schlüsselwert von s in Q auf 0}
$\pi[s] \leftarrow$ NIL
$S \leftarrow \emptyset$
while $Q \neq \emptyset$ **do**
  $u \leftarrow$ EXTRACT-MIN(Q)
  $S \leftarrow S \cup \{u\}$
  for all $v \in \mathrm{Adj}[u]$ **do**
    if $d[v] > d[u] + \ell(u, v)$ **then**
      DECREASE-KEY$(Q, v, d[u] + \ell(u, v))$ {Vermindere den Schlüssel $d[u]$ von u in Q auf $d[u] + \ell(u, v)$}
      $\pi[v] \leftarrow u$

(a) Beweisen Sie, daß bei Abbruch des obigen Algorithmus $d[u] = \delta(s, u)$ für alle $u \in V$ gilt. **(3 Punkte)**

(b) Zeigen Sie, daß man den Algorithmus so implementieren kann, daß er in Zeit $\mathcal{O}(|V| \log |V| + |E|)$ läuft. **(2 Punkte)**

Abgabetermin: Mittwoch, 18. Juni 1997, bis 11^{30} Uhr in den dafür vorgesehenen Kästen vor der Teilbibliothek Mathematik/Informatik