Algorithmen und Datenstrukturen
Musterlösung 12

Martin Avanzini <martin.avanzini@uibk.ac.at>
Thomas Bauereiß <thomas.bauereiss@uibk.ac.at>
Herbert Jordan <herbert@dps.uibk.ac.at>
René Thiemann <rene.thiemann@uibk.ac.at>

14. Juni, zur Besprechung am 21. Juni

Aufgabe 1) Kürzeste Wege

1. Dijkstra's: Nachfolgende Tabelle zeigt die zeitliche Abfolge der Werte $d[]$. Knoten mit minimalen $d[]$ sind jeweils unterstrichen.

A	B	C	D	E	F	G
$\underline{0}$	∞	∞	∞	∞	∞	∞
	$\underline{1}$	2	3	∞	∞	∞
		$\underline{2}$	2	5	∞	∞
			$\underline{2}$	5	5	∞
				5	$\underline{4}$	7
				$\underline{5}$		5
						$\underline{5}$

   Kürzester Weg von A nach G: A-B-D-F-G, $\delta(A, G) = 5$

2. Kürzester Weg von A nach G unter Ausnützung der Tatsache dass Graph azyklisch.

   (a) Schritt 1) Topologische Ordnung - z.B. A, B, C, D, E, F, G.

   (b) Schritt 2) Verarbeitung in topologischer Ordnung

				d			
u	A	B	C	D	E	F	G
-	0	∞	∞	∞	∞	∞	∞
A	0	$\underline{1}$	$\underline{2}$	$\underline{3}$	∞	∞	∞
B	0	1	2	$\underline{2}$	$\underline{5}$	∞	∞
C	0	1	2	2	5	$\underline{5}$	∞
D	0	1	2	2	5	$\underline{4}$	$\underline{7}$
E	0	1	2	2	5	4	$\underline{6}$
F	0	1	2	2	5	4	$\underline{5}$
G	0	1	2	2	5	4	5

Die Nachfolger von u sind jeweils unterstrichen. Der Vorteil dieses Algorithmus liegt in der geringeren asymptotischen Laufzeit $\mathcal{O}(|V|+|E|)$. Die Beschränkung auf azyklische Graphen ist in der Notwendigkeit der Berechnung einer topologischen Ordnung begründet.

Aufgabe 2) Kürzeste Wege in azyklischen Graphen Das Theorem kann für einen Graphen mit $V = \{v_1, v_2, \ldots, v_n\}$ per Induktion über die topologisch sortierte Sequenz $v_1, v_2, \ldots, v_n$ bewiesen werden.

1. Induktionsannahme: $\forall i : d[v_i] = \delta(s, v_i)$
2. Induktionsbeginn: $i = 1$ Für v_1 gilt entweder $v_1 = s$ oder $v_1 \neq s$.
   - Falls $v_1 = s$, so ergibt der Algorithmus $d[v_1] = d[s] = 0$ da $d[s]$ im 2. Schritt auf 0 gesetzt. Weiters besitzt s keine Vorgänger, da s an erster Stelle in der topologischen Ordnung steht. Daher wird $d[s]$ im 3. Schritt nicht mehr verändert. Da im Graphen G keine Zyklen, insbesondere keine negativen Zyklen, enthalten sind, gilt $\delta(s, s) = 0 = d[s] = d[v_1]$.
   - Falls $v_1 \neq s$, so ergibt der Algorithmus $d[v_1] = \infty$ da im 2. Schritt entsprechend initialisiert und mangels Vorgänger nicht weiter aktualisiert. Da $v_1 \neq s$ keine Vorgänger hat, existiert auch kein Weg von s nach v_1. Daher gilt $\delta(s, v_1) = \infty = d[v_1]$.
3. Induktionsschritt: $i \to i + 1$
   - Falls $v_{i+1} = s$ gilt $\forall 1 \leq j \leq i : d[v_j] = \infty$ aufgrund der IH und da aus der topologischen Sortierung folgt, dass kein Weg von $s = v_{i+1}$ nach v_j führt. Weiters wird im 2. Schritt $d[v_{i+1}] = d[s]$ mit 0 initialisiert. Im 3. Schritt wird $d[v_{i+1}]$ auf $min(d[v_{i+1}], min\{d[u] + w(u, v_{i+1}) | (u, v_{i+1}) \in E\}) = min(d[v_{i+1}], \infty) = d[v_{i+1}] = 0$ aktualisiert. Daher, am Ende gilt: $d[v_{i+1}] = d[s] = 0$. Da G azyklisch gilt auch $\delta(s, s) = 0$ for alle $s \in V$.
   - Falls $v_{i+1} \neq s$ gilt

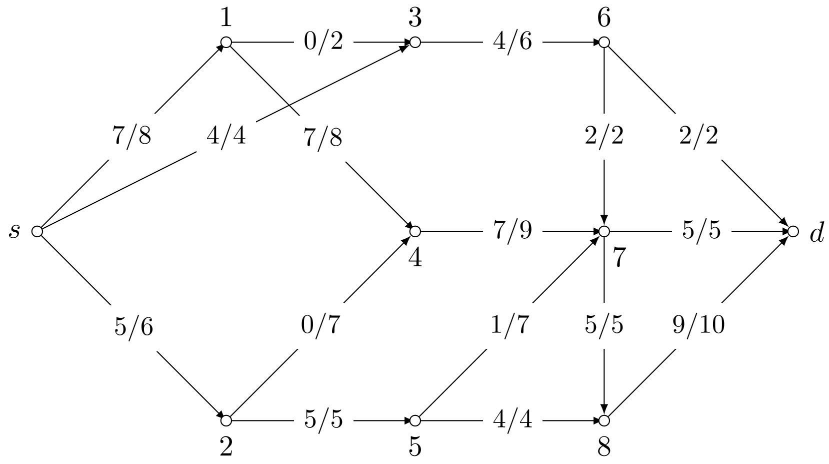
     $$\begin{aligned} \delta(s, v_{i+1}) &= min\{\delta(s, u) + w(u, v_{i+1}) | (u, v_{i+1}) \in E\} \\ &=^{IH} min\{d[u] + w(u, v_{i+1}) | (u, v_{i+1}) \in E\} \\ &= d[v_{i+1}] \end{aligned}$$

     da der Kürzeste Weg von s nach $v_{i+1} \neq s$ durch einen der Vorgänger von v_{i+1} verlaufen muss und alle Vorgänger in $v_1, \ldots, v_i$ enthalten sein müssen.

Aufgabe 3) Fluss-Netzwerke Einen maximalen Fluss erhält man mit den folgenden Erweiterungspfaden (die Zahlen über den Kanten geben an, um welchen Wert der Fluss erhöht bzw. erniedrigt wird):

- $s \xrightarrow{+5} 1 \xrightarrow{+5} 4 \xrightarrow{+5} 7 \xrightarrow{+5} 8 \xrightarrow{+5} d.$
- $s \xrightarrow{+2} 3 \xrightarrow{-2} 1 \xrightarrow{+2} 4 \xrightarrow{+2} 7 \xrightarrow{-2} 5 \xrightarrow{+2} 8 \xrightarrow{+2} d.$
- $s \xrightarrow{+2} 3 \xrightarrow{+2} 6 \xrightarrow{+2} 7 \xrightarrow{-2} 5 \xrightarrow{+2} 8 \xrightarrow{+2} d.$

Dadurch erhält man den folgenden maximalen Fluss:



Wenn man die Kapizität der Kante $(7, d)$ von 5 auf 7 erhöht, dann kann man den maximalen Fluss um 2 erhöhen:

- $s \xrightarrow{+1} 1 \xrightarrow{+1} 4 \xrightarrow{+1} 7 \xrightarrow{+1} d.$
- $s \xrightarrow{+1} 2 \xrightarrow{+1} 4 \xrightarrow{+1} 7 \xrightarrow{+1} d.$