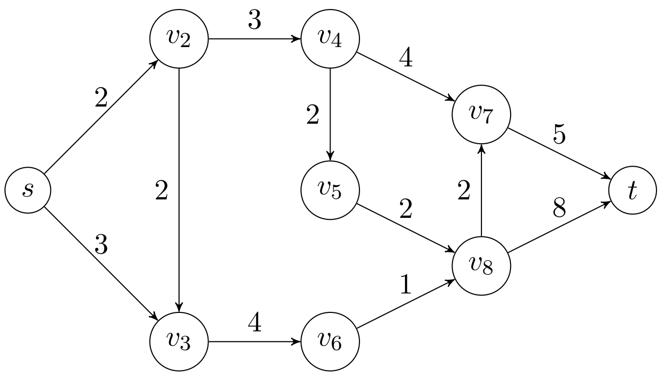
Technische Universität München, Zentrum Mathematik
Lehrstuhl für Angewandte Geometrie und Diskrete Mathematik

Algorithmische Diskrete Mathematik (MA 2501)

Prof. Dr. P. Gritzmann | Dr. M. Ritter | Dipl.-Math. M. Silbernagl

Aufgabenblatt 6

Tutoraufgabe 6.1 (Kürzeste Wege-Probleme)
Berechnen Sie mit Hilfe des Dijkstra-Algorithmus im untenstehenden Digraphen einen kürzesten s-t-Weg. Die Zahlen an den Kanten geben jeweils die Kantengewichte an. Stellen Sie Ihre Rechnung so dar, dass die Einzelschritte nachvollziehbar sind.



Aufgabe 6.2 (kürzeste Wege und kürzeste Kantenzüge) [5]
Sei $G = (V, E)$ ein Digraph, $s, t \in V$ zwei verschiedene Knoten in G und $w : E \to \mathbb{R}$ eine Kantengewichtung. Ferner seien $\mathcal{K}$, $\mathcal{E}$ und $\mathcal{W}$ die Menge aller Kantenzüge, aller einfachen Kantenzüge bzw. aller Wege von s nach t in G und φ die Zielfunktion einer Aufgabe kürzester Kantenzüge, kürzester einfacher Kantenzüge bzw. kürzester Wege von s nach t. Beweisen Sie die folgenden Aussagen:

a) Das Minimum von φ über $\mathcal{E}$ und das Minimum von φ über $\mathcal{W}$ existiert genau dann, wenn $\mathcal{W} \neq \emptyset$.

b) Das Minimum von φ über $\mathcal{K}$ existiert genau dann, wenn $\mathcal{K} \neq \emptyset$ und wenn für jeden Kantenzug $K \in \mathcal{K}$ gilt: Kein Knoten in K ist in einem geschlossenen Kantenzug negativer Länge enthalten.

c) Wird das Minimum von φ über $\mathcal{K}$ angenommen, so existiert ein s-t-Weg W höchstens gleicher minimaler Länge.

d) Ist $\mathcal{K} \neq \emptyset$ und $w_e > 0$ für alle $e \in E$, so ist jeder minimale Kantenzug von s nach t ein s-t-Weg.

Aufgabe 6.3 (Der Dijkstra-Algorithmus und negative Kantengewichte) [6]

a) Geben Sie ein Beispiel eines Digraphen mit negativen Kantengewichten, bei dem der Dijkstra-Algorithmus ein falsches Ergebnis liefert. Warum funktioniert der Korrektheitsbeweis nicht mehr, auf dem der Algorithmus basiert?

b) Um das Problem zu beheben, könnte man versuchen, auf alle Kantengewichte im Digraphen die gleiche positive Konstante $c \in \mathbb{N}$ zu addieren, um negative Kanten zu eliminieren. Warum ist ein vom Dijkstra-Algorithmus gefundener kürzester Weg im veränderten Digraphen im Allgemeinen nicht kürzester Pfad im unveränderten Digraphen? Illustrieren Sie das Problem anhand eines Beispiels.

Bitte wenden!

c) Um auch Digraphen mit negativen Kantengewichten zu behandeln, ändern wir die Update-Regel aus Prozedur 4.3.6 so ab, dass auch alle bereits in S enthaltenen Knoten nochmals überprüft werden, die erste Update-Schleife läuft also statt über $v \in N_{\text{aus}}(G, v^*) \setminus (S \cup N_{\text{aus}}(G, S))$ über $v \in N_{\text{aus}}(G, v^*) \setminus N_{\text{aus}}(G, S)$. Zeigen Sie, dass auch diese Änderung das Problem nicht behebt. Was ist von rekursiver Anwendung der Update-Regel zu halten (d. h. nach Aktualisierung eines Knoten-Labels aktualisiert man immer die Label aller Nachbarn)?

d) Sei G nun ein Digraph, der keinen Kreis negativer Länge enthält und in dem nur solche Kanten negative Gewichte haben dürfen, die im Startknoten s beginnen. Liefert der Dijkstra-Algorithmus nun korrekte Ergebnisse?

Aufgabe 6.4 (Dijkstra vs. Prim) [2]

Zeigen oder widerlegen Sie: Werden die Algorithmen von Dijkstra und Prim auf einem zusammenhängenden gewichteten Graphen $G = (V, E, c)$ mit Startknoten $v \in V$ aufgerufen, so liefern sie die gleichen Bäume, d.h. insbesondere liefert Dijkstra auch einen MST.
Der von Dijkstra erzeugte Baum enthält dabei für jeden Knoten $v \in V \setminus \{s\}$ die Kante $\{v^*, v\}$, wobei v^* der gewählte Knoten des Schleifendurchlaufs ist, in dem die Knotenmarke $\xi(v)$ zum letzten Mal aktualisiert wurde.

Aufgabe 6.5 (Bellman-Ford) [7]

Sei $G = (V, E, \phi)$ ein Digraph mit Kantengewichtung $\phi : E \to \mathbb{R}$ ohne negative Kreise und $s \in V$. Betrachten Sie folgenden Algorithmus zur Bestimmung der Längen der kürzesten s-v-Wege für alle $v \in V$:

Input: Digraph $G = (V, E, \phi)$ ohne negative Kreise, Startknoten $s \in V$
Output: $\xi : V \to \mathbb{R}$, wobei $\xi(v)$ die Länge eines kürzesten s-v-Weges ist.
$\xi^0(s) \leftarrow 0$;
for $v \in V \setminus \{s\}$ **do**
  $\xi^0(v) \leftarrow \infty$;
for $k \in [|V| - 1]$ **do**
  for $v \in V$ **do**
    $\xi^k(v) \leftarrow \min\left\{\xi^{k-1}(v), \min\limits_{(u,v)\in E}\left\{\xi^{k-1}(u) + \phi(u, v)\right\}\right\}$;
return $\xi = \xi^{|V|-1}$;

a) Beweisen Sie, dass der Algorithmus die Länge $\xi(v)$ eines kürzesten s-v-Weges für alle $v \in V$ bestimmt.

b) Erweitern Sie den Algorithmus so, dass die Existenz eines negativen Kreises in G entdeckt werden kann und zum Abbruch mit einer entsprechenden Fehlermeldung führt.

c) Bestimmen Sie die Laufzeit des Algorithmus.

Hinweis: Beweise Sie Teil a) mit Induktion über k:

1. $\xi^k(v)$ ist höchstens gleich der Länge eines kürzesten s-v-Weges mit nicht mehr als k Kanten, falls ein solcher existiert.
2. Ist $\xi^k(v) \neq \infty$, dann gibt $\xi^k(v)$ die Länge eines s-v-Weges an.

Die Abgabedaten von Blatt 6 für Ihre Gruppe finden Sie auf der Homepage zur Vorlesung unter `http://www-m9.ma.tum.de/WS2012/ADM`.