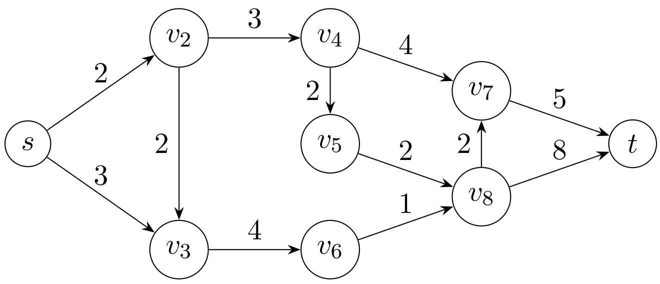
Algorithmische Diskrete Mathematik (MA 2501)

Prof. Dr. P. Gritzmann | Dr. M. Ritter | Dipl.-Math. F. Schmiedl

Aufgabenblatt 7

Tutoraufgabe 7.1 (Kürzeste Wege-Probleme) [9 Punkte]
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 7.2 (Der Dijkstra-Algorithmus und negative Kantengewichte) [7 Punkte]

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 negative Kantengewichte in positive zu transformieren, könnte man einfach eine ausreichend große, positive Konstante $c \in \mathbb{N}$ auf jedes Kantengewicht addieren. Lässt sich das Problem auf diese Weise beheben? Begründen Sie Ihre Antwort mit einem Beweis oder einem Gegenbeispiel.

c) 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 in solchen Graphen korrekte Ergebnisse?

Aufgabe 7.3 (Dijkstra vs. Prim) [3 Punkte]
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 der Dijkstra-Algorithmus einen MST.

Der von Dijkstra erzeugte Baum enthält dabei für jeden Knoten $v \in V \setminus \{s\}$ diejenige Kante, über die der Knotenmarker zuletzt geändert wurde.

Aufgabe 7.4 (Maximale Teilsumme) [6 Punkte]
Geben Sie einen Algorithmus mit Laufzeit $\mathcal{O}(n)$ an, der für jeden beliebigen Vektor $v \in \mathbb{R}^n$ den Wert einer maximalen zusammenhängenden Teilsumme findet, d. h. den Wert

$$\max_{a,b \in [n], a \le b} \sum_{i=a}^{b} v_i$$

berechnet. Beweisen Sie die Korrektheit Ihres Algorithmus und wenden Sie ihn auf den Vektor $v = (3, 5, -2, 4, 7, -8)^T$ an. *Hinweis: Verwenden Sie dynamische Optimierung und betrachten Sie das Teilproblem „Finde den Wert einer maximalen zusammenhängenden Teilsumme von v, welche in gegebenem b endet“.*

Bitte wenden!

Aufgabe 7.5 (Kürzester k-Weg) [8 Punkte]

Das folgende Problem bezeichnet man als KÜRZESTER k-WEG-Problem:

Eingabe: Ein Digraph $G = (V, E)$ mit einer Kantengewichtung $d : E \to \mathbb{N}$, eine natürliche Zahl $k \in \mathbb{N}$, ein Startknoten $s \in V$

Aufgabe: Bestimme für jeden Knoten $v \in V$ die Länge eines bezüglich d kürzesten s-v-Weges in G, der höchstens k Kanten verwendet, oder stelle fest, dass kein solcher Weg existiert.

Verwenden Sie dynamische Optimierung, um einen Algorithmus zu entwerfen, der KÜRZESTER k-WEG löst, indem Sie folgende Größen betrachten:

$$D(v, i) := \text{Länge eines kürzesten } s\text{-}v\text{-Weges, der } \textit{genau } i \text{ Kanten verwendet}$$

a) Ergänzen Sie (ohne Begründung) die folgende Rekursion für die $D(v, i)$ korrekt:

$$D(v, i) = \boxed{\min / \max} \left\{ D(u, i-1) + \boxed{} : (u, v) \in E \right\}$$

b) Ergänzen Sie den folgenden Algorithmus so, dass er KÜRZESTER k-WEG korrekt löst.

$D(s, 0) \leftarrow \boxed{}$

for $v \in V \setminus \{s\}$ **do**
  $D(v, 0) \leftarrow \boxed{}$

for $i \in \boxed{}$ **do**
  for $v \in \boxed{}$ **do**
    $D(v, i) \leftarrow \boxed{}$

return $\left(\boxed{} \right)_{v \in V}$

***Bonusaufgabe 7.6** (Der A^*-Algorithmus) [8 Punkte]

Sei $G = (V, E)$ ein zusammenhängender Graph mit Kantenlängen $\ell : E \to \mathbb{R}_{>0}$ und $s, t \in V$. Für zwei Knoten $u, v \in V$ sei $\operatorname{dist}(u, v)$ die Länge eines kürzesten u-v-Weges und $h : V \to \mathbb{R}_{\geq 0}$ eine (heuristisch gefundene) untere Schranke für die Distanz zu t, d. h. für alle Knoten $v \in V$ gilt $h(v) \leq \operatorname{dist}(v, t)$. Für alle Kanten $\{u, v\} \in E$ gelte $h(u) \leq h(v) + \ell(\{u, v\})$. Zeigen Sie, dass der folgende Algorithmus dann die Länge eines kürzesten s-t-Weges berechnet. *Hinweis: Zeigen Sie, dass $\xi(x) = \operatorname{dist}(s, x) + h(x)$ für jeden Knoten x, der aus der Menge Q entfernt wurde.*

Input: G, ℓ, s, t, h

for $x \in V \setminus \{s\}$ **do**
  $\xi(x) \leftarrow \infty$;
$\xi(s) \leftarrow h(s)$
$Q \leftarrow V$
while $Q \neq \emptyset$ **do**
  $x \leftarrow$ Knoten in Q, der $\xi(\cdot)$ minimiert
  if $x = t$ **then**
    return $\xi(x)$
  $Q \leftarrow Q \setminus \{x\}$
  for $y \in N(x) \cap Q$ **do**
    $\xi(y) \leftarrow \min\{\xi(y), \xi(x) + \ell(\{x, y\}) + h(y) - h(x)\}$

Die Abgabedaten von Blatt 7 für Ihre Gruppe finden Sie auf der Homepage zur Vorlesung unter `http://www-m9.ma.tum.de/WS2014/ADM`.