Institut für Theoretische Informatik
Lehrstuhl Prof. Dr. D. Wagner

Aufwärm-Übungsblatt 1 – Lösungsvorschläge

Algorithmen für Routenplanung, Sommer 2010

Problem 1: Eigenschaften kürzester Wege *

Gegeben sei ein einfacher, gewichteter und gerichteter Graph $G = (V, E)$ mit einer nicht-negativen Gewichtsfunktion $\text{len} : E \to \mathbb{R}_0^+$. Weiterhin sei zu zwei beliebigen Knoten $s, t \in V$ ein kürzester Weg $\Pi(s,t) = (v_1, v_2, \ldots, v_k)$ mit $v_1 = s$ und $v_k = t$ gegeben.

(a) Zeigen Sie: Teilwege von kürzesten Wegen sind selbst kürzeste Wege. Das heißt für zwei $v_i, v_j \in \Pi(s,t)$ mit $i < j$ gibt es einen kürzesten Weg $\Pi(v_i, v_j)$ mit $\Pi(v_i, v_j) \subseteq \Pi(s,t)$.

Lösung. Wir beweisen die Behauptung durch Widerspruch. Sei also $\Pi(s,t) = (v_1, v_2, \ldots, v_k)$ mit $v_1 = s$ und $v_k = t$ ein kürzester s-t-Weg. Weiterhin betrachte einen Teilweg $\Pi(v_i, v_j)$ mit $i < j$ von $\Pi(s,t)$ der selbst *kein* kürzester Weg ist. Das heißt, es gibt einen Weg $\Pi'(v_i, v_j)$ mit $\Pi'(v_i, v_j) \neq \Pi(v_i, v_j)$ und $\text{len}(\Pi'(v_i, v_j)) < \text{len}(\Pi(v_i, v_j))$. Substituieren wir in $\Pi(s,t)$ den Teilweg $\Pi(v_i, v_j)$ durch $\Pi'(v_i, v_j)$ erhalten wir einen s-t-Weg $\Pi'(s,t)$ mit $\text{len}(\Pi'(s,t)) < \text{len}(\Pi(s,t))$. Dies ist ein Widerspruch dazu, dass $\Pi(s,t)$ ein kürzester Weg war. □

(b) Zeigen Sie: Für zwei Knoten $s, t \in V$ gibt es immer einen *einfachen* kürzesten Weg $\Pi(s,t)$. Ein Weg $P = (v_1, \ldots, v_k)$ heißt einfach, wenn es keine zwei Indizes $1 \leq i, j \leq k$ in P mit $i \neq j$ gibt, so dass $v_i = v_j$.

Lösung. Die Argumentation verläuft ähnlich zu Aufgabe (a). Sei

$$\Pi(s,t) = (v_1, \ldots v_i, w, \ldots, w, v_j, \ldots, v_k)$$

ein kürzester s-t-Weg mit $v_1 = s$ und $v_k = t$ der einen Knoten w mehrfach enthält. Betrachte den Weg der aus $\Pi(s,t)$ hervorgeht indem wir den Kreis um w entfernen, also $\Pi'(s,t) := (v_1, \ldots, v_i, w, v_j, \ldots, v_k)$. Offensichtlichtlich ist $\Pi'(s,t)$ ein gültiger s-t-Weg. Weiterhin ist $\text{len}(\Pi'(s,t)) = \text{len}(\Pi(s,t)) - \text{len}(\Pi(w,w))$ wobei $\Pi(w,w)$ der Teilweg von $\Pi(s,t)$ ist, der aus dem Kreis um w besteht. Ist $\Pi(w,w) > 0$, so war $\Pi(s,t)$ kein kürzester Weg, was nicht sein kann. Ist hingegen $\Pi(w,w) = 0$, so ist $\text{len}(\Pi'(s,t)) = \text{len}(\Pi(s,t))$ und somit $\Pi'(s,t)$ wieder ein kürzester Weg. Dieser Schritt kann nun wiederholt werden bis wir einen einfachen s-t-Weg erhalten. □

Problem 2: Dijsktra's Algorithmus **

Gegeben sei ein einfacher, gewichteter und ungerichteter Graph $G = (V, E)$ mit einer nicht-negativen Gewichtsfunktion $\text{len} : E \to \mathbb{R}_0^+$.

(a) Beweisen Sie die Korrektheit das Stopkriteriums von Dijkstra's Algorithmus aus der Vorlesung.

Hinweis: Zeigen Sie dazu, dass zu einer s-t-Anfrage für jeden Knoten $v \in V$ nachdem er

abgearbeitet wurde, das Distanzlabel $d[v]$ bei Dijkstra's Algorithmus die tatsächliche Distanz $d(s, v)$ enthält.

Lösung. Es bezeichne $S \subset V$ die Menge abgearbeiteter Knoten (oft auch *Suchraum* genannt) und entsprechend $V \setminus S$ die Menge von Knoten, die noch nicht abgearbeitet wurden. Weiterhin sei angemerkt, dass Dijkstra's Algorithmus als nächsten abzuarbeiteten Knoten immer denjenigen Knoten $u \in Q$[1] mit $S \subset V \setminus S$ auswählt für den $d[u]$ minimal ist.

Nehmen wir nun an, dass Dijkstra's Algorithmus einen Knoten $u \in V$ abarbeitet, mit $d[u] > \text{dist}(s, u)$. Sei außerdem u der Knoten, bei dem dies zum *ersten* Mal auftritt. Da sich im Verlauf des Algorithmus die Werte $d[u]$ für alle $u \in V$ als Summe der Kantengewichte entlang eines Pfades von s nach u zusammensetzen, gilt stets

$$\text{dist}(s, u) \leq d[u] \qquad \forall u \in V,$$

da es keinen kürzeren Weg als den kürzesten Weg geben kann.

Sei $\Pi(s, u)$ der induzierte Pfad zu u. Da $\Pi(s, u)$ nicht der kürzeste Weg ist, gibt es einen weiteren Knoten $v \in V \setminus S$ mit $\text{dist}(s, v) < d[u]$ der noch nicht abgearbeitet wurde. Betrachten wir den kürzesten Weg zu v, also $\Pi(s, v)$. Da dieser bei s beginnt, und $s \in S$ aber $v \notin S$ gilt, gibt es einen ersten Knoten $y \notin S$ entlang des kürzesten Weges zu v. Sei $x \in S$ der Vorgängerknoten auf dem Pfad. Siehe auch Abbildung 1 zur Veranschaulichung.

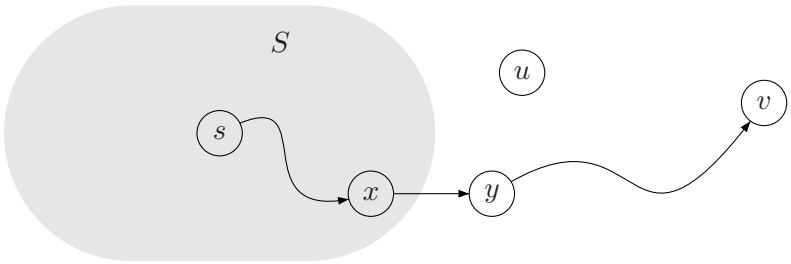


Abbildung 1: Situation im Beweis zu Aufgabe (a).

Es gilt also

$$\text{dist}(s, x) \leq \text{dist}(s, y) \leq \text{dist}(s, v).$$

Da u der erste inkorrekt abgearbeitete Knoten während des Ablaufs von Dijkstra's Algorithmus ist, gilt für x noch $\text{dist}(s, x) = d[x]$ und somit wurde zuvor, als x abgearbeitet wurde, in Dijkstra's Algorithmus $d[y] := \text{dist}(s, x) + \text{len}(x, y) = \text{dist}(s, y)$ gesetzt und y ggf. in Q eingefügt. Damit folgt insgesamt

$$d[y] = \text{dist}(s, y) \leq \text{dist}(s, v) < d[u]$$

und $y \in Q$. Dies ist jedoch ein Widerspruch zu der Annahme dass $u \in Q$ derjenige Knoten ist, der $d[u]$ minimiert (da man durch Wahl von y einen Knoten erhielte mit $d[y] < d[u]$).

Es gilt also für jeden abgearbeiteten Knoten $u \in V$ dass $d[u] = \text{dist}(s, u)$; insbesondere auch für t. Das Abbruchkriterium von Dijkstra's Algorithmus ist also korrekt. □

(b) Angenommen wir modifizieren Dijkstra's Algorithmus (ohne Stoppkriterium) so, dass zu einem Knoten $u \in V$ nicht länger alle Kanten $(u, v) \in E$ relaxiert werden, sondern lediglich eine geeignete Teilmenge der Kanten relaxiert wird.

[1] Q ist die Priority-Queue

Zeigen Sie: Zu jeder Anfrage gibt es eine Teilmenge $E' \subseteq E$ der zu relaxierenden Kanten, so dass Dijkstra's Algorithmus nur $|V| - 1$ Relaxierungsschritte durchführt, und trotzdem für alle Knoten $v \in V$ das Korrekte Ergebnis berechnet wird.

Lösung. Wir betrachten Dijkstra's Algorithmus ohne der Modifikation aus dieser Aufgabe. Da Dijkstra's Algorithmus korrekt ist, wird für jeden Knoten $u \in V$ die Distanz $d[u]$ korrekt berechnet, das heißt nach Ausführung von Dijkstra's Algorithmus gilt $d[u] = \text{dist}(s, u)$ für alle $u \in V$.

Bei der Ausführung von Dijkstra's Algorithmus wurde des Weiteren für jeden Knoten $v \in V \backslash \{s\}$ sein Vorgängerknoten $p[v]$ auf dem (vorläufig) kürzesten Weg zu v Berechnet. Nach Ausführung ist also für jeden Knoten $v \in V \setminus \{s\}$ die Kante $(p[v], v) \in E$ diejenige Kante entlang des kürzesten Weges, über die v erreicht wird. Definieren wir also

$$E' := \{(u, v) \in E \mid u = p[v]\}$$

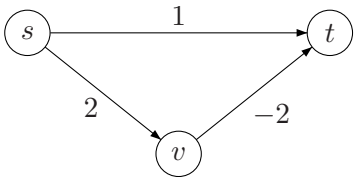
so folgt aus $\exists p[v] \in V$ für alle $V \backslash \{s\}$ dass $|E'| = |V| - 1$. Das heißt, würde Dijkstra's Algorithmus nur Kanten aus E' relaxieren, so würde dennoch für jeden Knoten $v \in V$ der kürzeste Weg zu v korrekt berechnet werden.

□

Sei nun len nicht länger nicht-negativ, das heißt es gilt len : $E \to \mathbb{R}$. Es gebe jedoch in G keine Kreise mit negativem Gewicht, das heißt für alle Wege $P = (v_1, \ldots, v_k)$ mit $v_1 = v_k$ ist $|P| \geq 0$.

(c) Geben Sie einen Graphen G und zwei ausgezeichnete Knoten $s, t \in V$ an, für die Dijkstra's Algorithmus nicht den kürzesten Weg findet.

Lösung. Wir betrachten folgenden Graphen:



Bei der Berechnung eines s-t-Weges wird nun zunächst Knoten t abgearbeitet mit $d[t] = 1$. Als nächstes wird Knoten v abgearbeitet mit $d[v] = 2$. Der kürzeste Weg von s nach t wäre aber (s, v, t) mit Gewicht 0. Dieser wird jedoch nicht gefunden. □

Problem 3: Distanzmetrik in Graphen *

Gegeben sei ein ungerichteter, gewichteter Graph $G = (V, E)$ mit einer nicht-negativen Gewichtsfunktion len : $E \to \mathbb{R}_0^+$.

(a) Zeigen Sie, dass die durch len induzierte Distanzfunktion $d : V \times V \to \mathbb{R}$ auf G folgende Eigenschaften hat:

(i) Nichtnegativität: $d(u, v) \geq 0$ für alle $u, v \in V$.

(ii) Dreiecksungleichung: $d(u, v) + d(v, w) \geq d(u, w)$ für alle $u, v, w \in V$.

(iii) Symmetrie: $d(u,v) = d(v,u)$ für alle $u, v \in V$ genau dann wenn G ungerichtet ist.

Lösung. Wir zeigen die Eigenschaften im Folgenden.

(i) Nichtnegativität.

Folgt unmittelbar aus $\text{len}(u,v) \geq 0$ für alle $u, v \in V$. Es kann daher keinen kürzesten Weg $\Pi(u,v)$ geben mit $\sum_{(x,y)\in\Pi(u,v)} \text{len}(x,y) < 0$.

(ii) Dreiecksungleichung.

Angenommen die Dreiecksungleichung gelte nicht. Das heißt es gibt Knoten $u, v, w \in V$ mit $d(u,v) + d(v,w) < d(u,w)$. Sei $\Pi(u,v)$ der durch $d(u,v)$ induzierte kürzeste Weg, $\Pi(v,w)$ der durch $d(v,w)$ induzierte kürzeste Weg und $\Pi(u,w)$ der durch $d(u,w)$ induzierte "kürzeste" Weg. Offenbar erhalten wir einen neuen kürzesten u-w-Weg $\Pi'(u,w)$ durch Konkatenation der Wege $\Pi(u,v)$ und $\Pi(v,w)$. Widerspruch.

(iii) Symmetrie.

Angenommen für zwei Knoten $u, v \in V$ ist $d(u,v) \neq d(v,u)$ und o. B. d. A. $d(u,v) < d(v,u)$. Sei $\Pi(v,u)$ der durch $d(v,u)$ induzierte Weg und $\Pi(u,v)$ der durch $d(u,v)$ induzierte Weg. Da G ungerichtet ist, erhalten wir einen kürzeren v-u-Weg, indem wir $\Pi(u,v)$ invertieren. Widerspruch.

□

(b) Sei nun G ein *gerichteter* Graph. Konstruieren Sie basierend auf d eine neue Distanzfunktion $d' : V \times V \to \mathbb{R}$, die alle Eigenschaften aus Aufgabe (a) erfüllt.

Lösung. Wir konstruieren eine neue Distanzfunktion $d' : V \times V \to \mathbb{R}$, die auf d basiert wie folgt:

$$d'(u,v) := \frac{1}{2}\big(d(u,v) + d(v,u)\big) \qquad \forall u, v \in V.$$

Offensichtlich erfüllt diese Distanzfunktion die Symmetrieeigenschaft. Weiterhin wird wegen $d(u,v) \geq 0$ und $d(v,u) \geq 0$ auch die Nichtnegativitätseigenschaft erfüllt. Zu überprüfen bleibt die Dreiecksungleichung.

Nachrechnen liefert

$$\begin{aligned} d'(u,v) + d'(v,u) &\geq d'(u,w) \\ \frac{1}{2}\big(d(u,v) + d(v,u)\big) + \frac{1}{2}\big(d(v,w) + d(w,v)\big) &\geq \frac{1}{2}\big(d(u,w) + d(w,u)\big) \\ d(u,v) + d(v,u) + d(v,w) + d(w,v) &\geq d(u,w) + d(w,u) \\ \big(d(u,v) + d(v,w)\big) + \big(d(w,v) + d(v,u)\big) &\geq d(u,w) + d(w,u) \end{aligned}$$

Dass diese Gleichung gilt folgt unmittelbar aus der Dreiecksungleichung für d, da gilt

$$d(u,v) + d(v,w) \geq d(u,w) \qquad \text{und} \qquad d(w,v) + d(v,u) \geq d(w,u)$$

□