

Übungsblatt 11

Abgabe: Dienstag, 29. Juli, 16:00 Uhr

Dieses Übungsblatt gibt 30 Punkte.

Um für die Klausur zugelassen zu sein brauchen Sie 125 der insgesamt 250 Punkte.

Aufgabe 1 (10 Punkte)

Betrachten Sie den Algorithmus 1, *Fast-Bellman-Ford*, welcher als Eingabe einen *gerichteten, gewichteten* und *kreisfreien* Graphen $G = (V, E, w)$ sowie $s \in V$ erhält (w ist hierbei die Gewichtsfunktion $w : E \mapsto \mathbb{R}$, mit erlaubten negativen Gewichten).

Algorithm 1 Fast-Bellman-Ford

Sortiere V *topologisch* und speichere die sortierte Liste im Array B ab.

for all $v \in V$ **do**

$\delta(s, v) \leftarrow \infty$

$parent(v) \leftarrow \text{NIL}$

$\delta(s, s) = 0$; $i = 1$

while $i \leq n$ **do**

$u \leftarrow B[i]$

for all out-neighbors v of u **do**

if $\delta(s, u) + w(u, v) < \delta(s, v)$ **then**

$\delta(s, v) = \delta(s, u) + w(u, v)$

$i \leftarrow i + 1$

Welche Laufzeit hat der Algorithmus? Begründen Sie Ihre Antwort. Argumentieren Sie zudem, dass der Algorithmus tatsächlich alle kürzesten Pfade berechnet.

Hinweis: Überlegen Sie, inwieweit die topologische Sortierung hilft und welche Invariante der Algorithmus dadurch aufrechterhält.

Aufgabe 2 (5 Punkte)

Gegeben sei ein *gewichteter, gerichteter* Graph $G = (V, E)$, in welchem wieder negative Kantengewichte zugelassen sind, aber keine Kreise negativen Gewichts existieren. Während die Länge eines Pfades in einem gewichteten Graph die Summe der Kantengewichte ist, so nennen wir die Anzahl der Kanten in einem Pfad seine *Hop-Länge* $h_l(P)$ (d.h., die Länge des Pfades, wenn G ungewichtet wäre). Sei s ein Knoten aus V . Ein *kürzester kürzester Pfad* von s zu v ist ein kürzester Pfad von s nach v mit minimaler Hop-Länge, d.h., gibt es mehrere kürzeste Pfade, so wird ein Unentschieden via Hop-Länge gebrochen; es kann natürlich mehrere kürzeste kürzeste Pfade geben). Außerdem definieren wir

$$\text{HOP-STRETCH}(s) := \max_{v \in V \setminus \{s\}} \{h_l(P) : P \text{ ist ein kürzester kürzester Pfad von } s \text{ nach } v\}$$

d.h., der Hop-Stretch von s ist das Maximum solcher kürzester kürzester Pfade zu/von s .

Wie müssen Sie den Bellman-Ford Algorithmus anpassen, so dass Sie statt der n Wiederholungen nur $hs(s) := \text{HOP-STRETCH}(s)$ viele Wiederholungen benötigen? Beantworten Sie die Frage zunächst, indem Sie $hs(s)$ als

Eingabeparameter betrachten. Erläutern Sie anschließend, wie Sie Ihren Algorithmus abändern können, so dass $hs(s)$ nicht bekannt sein muss, d.h., nicht übergeben werden muss, und der Algorithmus dennoch nach höchstens $hs(s) + 1$ Wiederholungen terminiert.

Aufgabe 3 (15 Punkte)

Füllen Sie den Online-Evaluationsbogen zur Lehrveranstaltung aus (Sie sollten dazu eine E-Mail erhalten haben). Insbesondere sind für uns dabei die geschriebenen Kommentare relevant, um die Lehrveranstaltung für zukünftige Studenten zu verbessern. Die Umfrage ist selbstverständlich anonym, schreiben Sie einfach in Ihre *erfahrungen.txt* für dieses Übungsblatt, dass Sie die Evaluation ausgefüllt haben. Das reicht uns als Beleg. Versuchen Sie bei Ihrer Kritik möglichst ehrlich und konstruktiv zu sein.

Wenn Sie *nicht* zur Klausur zugelassen sein wollen, das Durchführen der Evaluation Sie allerdings über die Punktegrenze bringen würde, so können Sie die Vorlesung natürlich auch evaluieren und nichts in die *erfahrungen.txt* Datei eintragen.