

9. Übung:

Algorithmen und Datenstrukturen

Sommersemester 2010

28. Juni 2010

Abgabe bis Montag, 05. Juli 2010, 10:15 im Briefkasten vor H426

Aufgabe 9.1: (Punkte 6)

Sei $G = (V, E)$ ein gerichteter Graph. Den *reversen* Graphen $G_r = (V, E_r)$ erhält man durch Umdrehen aller Kanten, d.h. $E_r = \{ (w, v) \mid (v, w) \in E \}$

- Geben Sie einen Algorithmus an, der aus der Adjazenzlistendarstellung von G , die Adjazenzlistendarstellung von G_r berechnet.
- Wie kann man aus einer topologischen Sortierung ord für G eine topologische Sortierung ord_r des reversen Graphen G_r machen ?

Aufgabe 9.2: (Punkte 6)

In der Vorlesung wurde ein Algorithmus zum topologischen Sortieren vorgestellt, der jeweils einen Knoten mit *Eingangsgrad* 0 (konzeptuell) aus dem Graphen entfernt und ihm die nächste Nummer zuordnet. Entwickeln Sie einen Algorithmus zum topologischen Sortieren, der stattdessen die Knoten mit *Ausgangsgrad* 0 betrachtet.

Aufgabe 9.3: (Punkte 8)

Sei $G = (V, E)$ ein gerichteter Graph. Benutzen Sie Sortieren durch Fachverteilung (Bucketsort), um für jede Kante $e = (v, w) \in E$, die Gegenkante $rev(e)$ zu berechnen, falls diese existiert. Genauer, berechnen Sie für jede Kante $e = (v, w)$

$$rev(e) = \begin{cases} (w, v), & \text{falls } (w, v) \in E \\ \text{NULL}, & \text{sonst} \end{cases}$$

Hinweis:

Nehmen Sie an, dass $V = \{ 1, \dots, n \}$ und erzeugen Sie zwei Listen E_1 und E_2 die jeweils alle Kanten enthalten. Sortieren Sie nun die Paare in E_1 lexikographisch nach (source, target) und die in E_2 lexikographisch nach (target,source) und überlegen Sie sich, wo jeweils die beiden Kanten (v, w) und (w, v) in den sortierten Listen landen.