

10. Übung:

## Algorithmen und Datenstrukturen

Sommersemester 2008

18. Juni 2008

---

Abgabe bis Donnerstag, 26. Juni 2008, vor der Übung

### Aufgabe 10.1: (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.  $E_r = \{ (w, v) \mid (v, w) \in E \}$

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

### Aufgabe 10.2: (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 10.3: (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.