Fachbereich IV - Abteilung Informatik
Algorithmen und Datenstrukturen
Universität Trier

Daniel Schmitt Tel.: (0651)201-3272

2. Übung:

Algorithmen und Komplexität

Wintersemester 2008 4. November 2008

Abgabe bis Montag, 17. November 2008, vor der Übung

Aufgabe 2.1: (8 Punkte)

Seien T, F, B, C die Partition der Kantenmenge nach einem DFS-Lauf und *dfsnum* und *compnum* die entsprechenden Nummerierungen der Knoten. Zeigen Sie:

a) es existiert ein Baumpfad von v nach w gdw. $dfsnum(v) \leq dfsnum(w)$ und $compnum(v) \geq compnum(w)$

b) für eine Kante (v, w) gilt $(v, w) \in T \cup F$ gdw. $dfsnum(v) < dfsnum(w)$

c) für eine Kante (v, w) gilt $(v, w \in B$ gdw. $dfsnum(v) \geq dfsnum(w)$ und $copnum(v) \leq compnum(w)$

d) für eine Kante (v, w) gilt $(v, w) \in C$ gdw. $dfsnum(v) > dfsnum(w)$ und $compnum(v) < compnum(w)$

Aufgabe 2.2: (8 Punkte)

a) Zeigen Sie dass fuer jeden DFS-Lauf auf einem azyklischen Graphen gilt: $compnum(v) > compnum(w)$

b) Verwenden Sie Teil a) zur Entwicklung eines effizienten Algorithmus zur Berechnung einer topologischen Sortierung.

Aufgabe 2.3: (4 Punkte)

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 $e = (v, w)$

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

Hinweis:
Nehmen Sie an, dass $V = \{ 1, \ldots, n \}$ 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.

Aufgabe 2.4: (5 Punkte)

Ein Graph $G = (V, E)$ heißt *ungerichtet*, wenn für jede Kante $(v, w) \in E$ auch $(w, v) \in E$ gilt. Im Allgemeinen zerfällt ein ungerichteter Graph in mehrere nicht miteinander verbundene Teile (Zusammenhangskomponenten). Verwenden und erweitern Sie DFS, um diese Teile zu berechnen. Genauer, wenn G aus ℓ Zusammenhangskomponenten $K_1, \ldots, K_\ell$ besteht, dann berechnen Sie für jeden Knoten v eine Zahl $num[v] \in \{ 1, \ldots, \ell \}$ mit $num[v] = i$ genau dann, wenn $v \in K_i$. *Hinweis:* Rufen Sie im Hauptprogramm `dfs(v)` für jeden noch nocht zuvor besuchten Knoten v auf und erweitern sie die Funktion so, dass die gesuchten Knotennummern zugewiesen werden.