

3. Übung:

Algorithmen und Komplexität

Wintersemester 2008

21. November 2008

Abgabe bis Montag, 24. November 2008, vor der Übung

Aufgabe 3.1:

(6 Punkte)

Zeigen Sie, dass für den in der Vorlesung behandelten Algorithmus zur Berechnung der starken Zusammenhangskomponenten stets folgende Invarianten erfüllt sind:

- a) Es existiert keine Kante $(v, w) \in E$ mit v in abgeschlossener SZK und w in nicht-abgeschlossener SZK.
- b) Alle nicht-abgeschlossenen SZK liegen auf einem Pfad, genauer: alle Wurzeln von nicht abgeschlossenen SZK liegen auf einem Baumpfad.
- c) Die Knoten jeder nicht-abgeschlossenen SZK bilden ein Intervall der Folge *unfertig*.

Aufgabe 3.2:

(8 Punkte)

Sei $G = (V, E)$ ein gerichteter Graph, betrachten Sie folgenden Algorithmus.

- a) Berechne eine completion-Nummerierung (*compnum*) von G mit DFS.
- b) Drehe alle Kanten in G um
- c) Markiere alle Knoten als nicht besucht.
- d) Durchlaufe die Knoten in absteigender *compnum*-Reihenfolge und rufe für jeden noch nicht besuchten Knoten v die rekursive Funktion ($dfs(v)$) auf.

Zeigen Sie, dass jeder Aufruf von *dfs* in Zeile 3 des Algorithmus genau eine starke Zusammenhangskomponente von G durchläuft.

Aufgabe 3.3:

(6 Punkte)

Ein Graph $G = (V, E)$ heißt *bipartit*, wenn man V in zwei disjunkte Teilmengen A und B zerlegen kann, so daß $E \subseteq A \times B \cup B \times A$, d.h. jede Kante führt von einem Knoten aus A zu einem Knoten aus B oder umgekehrt.

Entwickeln Sie einen effizienten Algorithmus, der testet, ob ein gegebener Graph bipartit ist.

Hinweis: Verwenden Sie BFS (Breitensuche).

Aufgabe 3.4:

(5 Punkte)

Die *transitive Hülle* eines Graphen $G = (V, E)$ ist der Graph $G' = (V, E')$ mit $(v, w) \in E'$ genau dann wenn ein Pfad von v nach w in G existiert. Entwickeln Sie einen Algorithmus zur Berechnung der transitiven Hülle.