

2. Übung:

Algorithmen und Komplexität

Wintersemester 2011

30. Oktober 2012

Abgabe bis

Aufgabe 2.1:

(10 Punkte)

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

1. Berechne eine completion-Nummerierung (*compnum*) von G mit DFS.
2. Drehe alle Kanten in G um
3. Markiere alle Knoten als nicht besucht.
4. 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 4 des Algorithmus genau eine starke Zusammenhangskomponente von G durchläuft.

Aufgabe 2.2:

(10 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).