

Klausur-Übungsblatt

Abgabetermin: 6.8.99

(4 Punkte)

1. Der Topological-Sort-Algorithmus, den wir in der Vorlesung diskutiert haben, findet eine von eventuell mehreren Lösungen. Beweisen Sie, daß ein Graph G existiert, so daß, unabhängig von der verwendeten Adjazenzliste, eine der möglichen Lösungen mit unserem Algorithmus nicht gefunden werden kann.

(4 Punkte)

2. Zeigen Sie, daß ein Graph existiert, für den zwei verschiedene Darstellungen der Adjazenzliste zur gleicher Reihenfolge der Knoten nach dem topologischen Sortieren führen.

(2 Punkte)

3. Warum kann man topologisches Sortieren nicht sinnvoll verwenden bei gerichteten Graphen, die Zyklen enthalten?

(20 Punkte)

4. Entwerfen Sie einen Algorithmus, der entscheiden kann, ob ein gegebener gerichteter Graph $G = (V, E)$ einen Zyklus beinhaltet oder nicht.

a) Schreiben Sie Ihren Algorithmus in Pseudocode.

b) Beschreiben Sie die Datenstrukturen, die Sie in Ihrem Algorithmus verwenden.

c) Implementieren Sie Ihren Algorithmus in Java.

d) Testen Sie Ihre Implementierung.

e) Machen Sie eine Zeitaufwand- und Speicheraufwand- Analyse mit Hilfe der O-Notation.

(8 Punkte)

5. Ein einfacher Algorithmus, um topologisches Sortieren zu implementieren, sucht wiederholt die Knoten mit Eingangsgrad gleich Null, und entfernt die jeweiligen Knoten zusammen mit allen ihren Ausgangskanten aus dem Graphen. Wie könnte man diesen Algorithmus implementieren, so daß man einen Zeitaufwand $O(V + E)$ hat? Schreiben Sie den entsprechenden Pseudocode.