

Übungen zu Algorithmen und Datenstrukturen Aufgabenblatt 10

Prof. Dr. Steffen Staab,
Dipl.–Inform. Gerd Gröner und Dipl.–Inform. Christoph Ringelstein
Arbeitsgruppe ISWeb, Fachbereich Informatik, Universität Koblenz–Landau

Abgabe: 28.01.2009, bis 23:59 Uhr

1 Graphen (8 Punkte)

Implementieren Sie die Klasse `KnotenListeGraph`, die einen gerichteten Graphen als Knotenliste darstellt. Für die Klasse sollen folgende Methoden implementiert werden:

- Implementieren sie die Member-Methode `int addNode()`, die einen Knoten in den Graph einfügt und die Anzahl der Knoten nach dem Einfügen zurückgibt.
- Implementieren Sie die Member-Methode `boolean addEdge(int orig, int dest)`, die eine Kante von Knoten `orig` nach Knoten `dest` einfügt. Die Methode gibt `false` zurück, falls mindestens einer der angegebenen Knoten nicht existiert oder die Kante bereits vorhanden ist.
- Implementieren Sie die Member-Methode `boolean deleteNode(int node)`, die einen Knoten aus dem Graphen löscht. Es werden alle eingehenden und ausgehenden Kanten mitgelöscht. Falls der Knoten nicht existiert, bleibt der Graph unverändert und der Rückgabewert ist `false`.
- Implementieren Sie die Member-Methode `boolean deleteEdge(int orig, int dest)` die eine Kante von Knoten `orig` zu Knoten `dest` löscht. Der Graph bleibt unverändert, falls die Kante nicht existiert. Der Rückgabewert ist in diesem Fall `false`, ansonsten `true`.

Die Knotenliste kann entweder als Array oder als verkettete Liste realisiert werden.

2 Graphalgorithmen (4 Punkte)

Implementieren Sie einen Breitendurchlauf für einen ungerichteten Graphen $G = (V, E)$ und einen Startknoten $s \in V$, für eine beliebige Anzahl von Knoten $n = \|V\|$. Das Ergebnis ist eine Menge von Kanten, die den aufspannenden Baum des Graphen darstellen. Diese Kanten sind die Kanten von jedem Knoten $v \in V \setminus \{s\}$ zu seinem Vorgängerknoten bei dem Breitendurchlauf. Der ungerichtete Graph ist als Adjazenzmatrix dargestellt.

- Eingabe: Adjazenzmatrix der Dimension $n \in \mathbb{N}$, Startknoten $s \in \{1, \dots, n\}$.
- Ausgabe: Aufspannender Baum E' , mit $E' \subseteq E \subseteq V \times V$

Wichtige Hinweise:

- Die SVN-Repositories sind unter <http://lehre2.isweb.uni-koblenz.de/svn/aud-ws0809/> zu erreichen, gefolgt vom jeweiligen Gruppennamen.
- Bitte checken Sie Ihr Programm und Ihre Texte und Grafiken ins SVN-Repository ein. Nutzen Sie als Verzeichnisname direkt unterhalb Ihres Gruppenverzeichnisses *zwingend*:

`blatt10`

Nur so können die Korrektoren Ihre Implementierung aus dem SVN-Repository auslesen und die Punkte verteilen.

- Achten Sie bitte darauf, dass Ihr Programm fehlerfrei übersetzt werden kann. Die Korrektoren werden Ihren Quellcode ebenfalls übersetzen und ausführen.
- Es werden nur Programme berücksichtigt, die in Java implementiert sind und direkt kompiliert werden können.
- Programme müssen gut kommentiert sein.
- **Bitte checken Sie KEINE .class-Dateien ein!**
- Wenn Sie Grafiken oder Textdokumente mit abgeben möchten, so verwenden Sie bitte als Dateiformat:
 - Für schematische Zeichnungen: .png oder .gif
 - Für Scans oder Photos: .png oder .jpg
 - Für Text: Plaintext (.txt) oder PDF-Dateien (.pdf). Natürlich dürfen Sie die Grafiken auch in eine PDF-Datei einbetten.

Andere Formate werden bei der Korrektur *nicht* berücksichtigt!

- Verwenden Sie bitte keine Umlaute, Leerzeichen oder Sonderzeichen in Dateinamen (auch nicht bei .pdf oder .png Dateien)!
- Verwenden Sie UTF-8 als Kodierung für Plaintext (.txt) und Java (.java) Dateien.