

Einführung in die Diskrete Mathematik

Programmieraufgaben

In diesen Programmieraufgaben sollen verschiedene Graphenalgorithmien implementiert werden. Die Lösungsprogramme sollen in der Lage sein, ein Kantenfile einzulesen, welches folgendes Format hat:

$\langle \text{AnzahlKnoten} \rangle \langle \text{AnzahlKanten} \rangle$
 $\langle \text{Knoten1} \rangle \langle \text{Knoten2} \rangle \langle \text{Gewicht} \rangle$

...

Beispiel eines Graphen mit 5 Knoten und 7 Kanten.

```
5 7
0 1 3
1 2 6
3 1 1
2 3 5
4 0 11
1 4 8
2 0 1
```

Auf diese Weise werden gerichtete Graphen kodiert, wobei die Kanten von $\langle \text{Knoten1} \rangle$ nach $\langle \text{Knoten2} \rangle$ gerichtet sein sollen. Wir betrachten stets nur schleifenfreie einfache Graphen ohne Paare gegenläufiger Kanten. Für Aufgabe 2, in der wir ungerichtete Graphen benötigen, kann man einfach zum zugrundeliegenden ungerichteten Graphen übergehen.

Testinstanzen für die Aufgaben befinden sich auf der Internetseite

http://www.or.uni-bonn.de/lectures/ws07/DM_uebung.html

Die Programme müssen in C, C++ oder Java geschrieben sein.

bitte wenden

Die Aufgaben:

Die Zahl der Knoten sei im folgenden mit n , die der Kanten mit m bezeichnet.

1. Schreibe ein Programm, das in Zeit $O(n+m)$ entscheidet, ob ein gegebener gerichteter Graph G einen (gerichteten) Kreis enthält. Wenn der Graph einen Kreis enthält, soll das Programm einen Kreis ausgeben. (8 Punkte)
2. Implementiere Borůvkas Algorithmus zur Berechnung eines minimalen Spannbaumes in einem ungerichteten Graphen. Das Programm soll Laufzeit $O(m \log n)$ haben. (8 Punkte)
3. Implementiere Yens Variante des Bellmann-Ford-Moore-Algorithmus. Das Verfahren soll auf allen Instanzen mit konservativen Kantengewichten funktionieren und in Zeit $O(nm)$ laufen. Der Startknoten soll der Knoten mit kleinstem Index sein (8 Punkte)

Abgabe: Spätestens bis Donnerstag, 10.1.2008. Wer seine Lösung bis zum 7.1.2008, 20 Uhr abgibt, erhält von seinem Tutor am 8.1. eine Rückmeldung über Kompilierbarkeit und grundsätzliche Fehler, so daß bis zum 10.1. eventuell noch eine verbesserte Version abgegeben werden kann. Die Lösungen bitte in jedem Fall per E-Mail direkt an den jeweiligen Tutor schicken.

Hinweis: Dieser Zettel ist der einzige Übungszettel mit Programmieraufgaben in diesem Semester. Seine erfolgreiche Bearbeitung (in der mindestens 50 % der möglichen Punkte erreicht werden müssen) ist eine notwendige Voraussetzung für den Übungsschein bzw. die Zulassung zur Modulprüfung.