Lehrstuhl Algorithmen
und Komplexität
Prof. Dr. Fabian Kuhn
Sebastian Daum, Oleksii Saukh

Algorithmen und Datenstrukturen
(Informatik II) SS 2014

http://ac.informatik.uni-freiburg.de/teaching

Übungsblatt 9

Abgabe: Dienstag, 15. Juli, 16:00 Uhr

Laden Sie Ihre Lösungen für theoretische *Übungen im Format "loesung-xx-y.pdf" hoch, wobei "xx" die Nummer des aktuellen Übungsblattes ist (bei Bedarf mit führender Null) und "y" die Aufgabennummer, d.h., erzeugen Sie* pro (theoretische) Aufgabe *eine* .pdf-*Datei. Scans, Bilder von Scans etc. bitte zuerst ins pdf-Format umwandeln. Bei Programmieraufgaben müssen Sie – wann immer es relevant/durchführbar ist – Unit Tests sowie einen Style-check durchführen. Laden Sie außerdem eine Datei "erfahrungen.txt" hoch, in welcher Sie Ihre Erfahrungen und Meinungen zu dem Übungsblatt teilen.*

Aufgabe 1 (4 Punkte)

Unter einem *zweitbesten MST* versteht man einen Spannbaum T, welcher von allen Spannbäumen auf dem Graphen G das zweitkleinste Gewicht erzielt.

In der Vorlesung wurde erwähnt, dass bei *paarweise unterschiedlichen* Kantengewichten der MST eindeutig ist, d.h., $w(e_i) \neq w(e_j)$ für je zwei beliebige Kanten e_i und e_j mit $i \neq j$. Beweisen oder widerlegen Sie:
Der zweitbeste MST ist in einem Graphen mit eindeutigen Kantengewichten eindeutig.

Aufgabe 2 (8 Punkte)

Sei G ein *gewichteter* und *zusammenhängender* Graph, in welchem alle Kantengewichte paarweise voneinander unterschiedlich sind. Nehmen Sie an, es existiert für jede Kante $e \in E$ die Information, ob diese Kante in *irgendeinem* Kreis im Graphen die schwerste Kante ist. Ihnen wird folgender *paralleler* Algorithmus präsentiert: Bei jeder Kante wird *simultan* nachgeprüft, ob sie die schwerste Kante in irgendeinem Kreis ist, und alle Kanten, auf die das zutrifft, werden *simultan* aus dem Graphen entfernt. *Beweisen* Sie, dass dieser Algorithmus den eindeutigen MST T hinterlässt.

Hinweis: Wenn Sie einen MST $T \subset G$ und einen Schnitt haben, so dass genau eine Kante e des MSTs über diesen Schnitt geht, und weiterhin e Teil eines Kreises $C \subset G$ ist, dann gibt es mindestens eine Kante f in C, welche auch über den Schnitt geht. (Diese Behauptung ist zwar offensichtlich, kann aber beim Lösen hilfreich sein, im Hinterkopf zu haben.)

Aufgabe 3 (8 Punkte)

Sie haben in der Vorlesung gesehen, wie man mit DFS einen gerichteten kreisfreien Graphen topologisch sortieren kann. Eine Variante zu DFS ist folgender Algorithmus. Man wählt einen Knoten ohne eingehende Kanten, fügt die ID an eine leere Liste an (welche später die topologisch sortierte Liste sein soll) und entfernt dann diesen Knoten und alle seine ausgehenden Kanten aus dem Graphen. Dies wiederholt man so lange, bis der Graph leer ist.

Gehen Sie davon aus, dass die Adjazenzlisten wie die aus der Vorlesung für gerichtete Graphen abgespeichert sind, d.h., existiert eine Kante von i nach j, dann ist ein Zeiger auf j in i's Adjazenzliste. Finden Sie eine Implementierung dieses Algorithmus, welcher in Laufzeit $O(n+m)$ läuft und begründen sie kurz, wieso er das Richtige tut und die Laufzeit korrekt ist.