Abteilung Algorithmik
Institut für Betriebssysteme und Rechnerverbund **SoSe 10**
TU Braunschweig

Nils Schweer
Dr. Alexander Kröller

Netzwerkalgorithmen
Übung 1 vom 14.04.2010

Abgabe der Lösungen bis Mittwoch, den 28.04.10, bis 13:00 Uhr in der Abteilung *Algorithmik* (IZ 262).
Bitte die Blätter vorne deutlich mit eigenem Namen und Gruppennummer versehen!

Aufgabe 1 (Wälder):
Ein Wald ist ein kreisfreier (nicht notwendigerweise zusammenhängender) Graph, d.h. ein Wald besteht aus einem oder mehreren Bäumen.
Beweise den folgenden Hilfssatz aus der Vorlesung: Sei W ein Wald mit n Knoten, m Kanten und p Zusammenhangskomponenten. Zeige mit Induktion über m, dass $n = m + p$ erfüllt ist.

(15 Punkte)

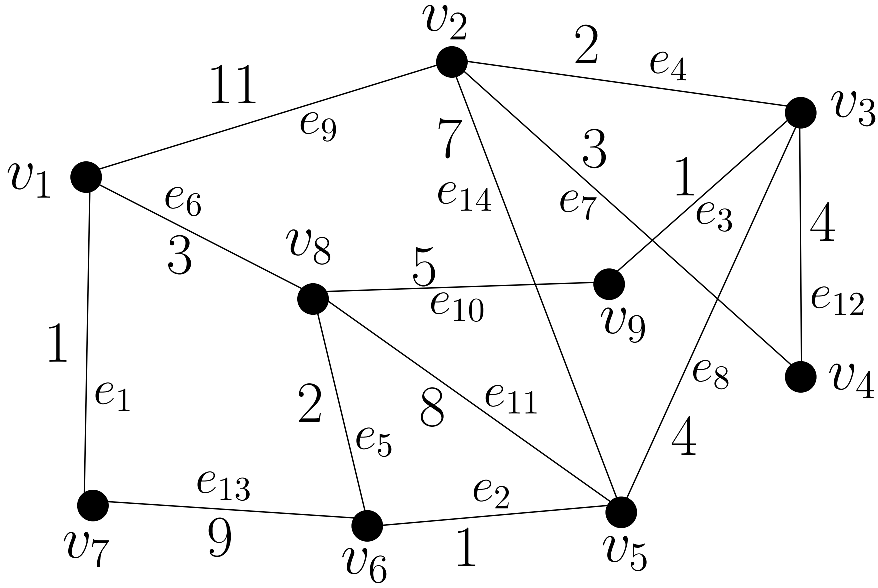
Aufgabe 2 (Divide and Conquer):
Professor Axtimwalde schlägt den folgenden Divide-and-Conquer-Algorithmus zur Bestimmung eines minimalen aufspannenden Baumes vor:
Für einen gegebenen zusammenhängenden Graphen $G = (V, E)$ wird die Knotenmenge V so in zwei Mengen V_1 und V_2 aufgespalten, dass sich ihre Kardinalitäten um höchstens eins unterscheiden. Sei E_1 die Menge der Kanten, die nur zu Knoten aus V_1 inzident sind und E_2 die Menge der Kanten, die nur zu Knoten aus V_2 inzident sind. Rekursiv wird nun das Minimum-Spanning-Tree-Problem für jeden der beiden Teilgraphen $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$ gelöst. Schließlich wird eine Kante von E mit minimalem Gewicht gewählt, für die ein Endknoten in V_1 und der andere in V_2 liegt, um die beiden resultierenden Bäume zu verbinden.
Funktioniert das immer? Zeige entweder, dass der Algorithmus immer einen minimalen aufspannenden Baum für G bestimmt, oder gib ein Beispiel an, in dem er es nicht schafft. (Welche Laufzeit hätte der Algorithmus?)

(15 Punkte)

Aufgabe 3 (Satz 2.6: Eigenschaften von Bäumen):
Zeige, dass es in einem Baum (kreisfreier und zusammenhängender Graph) einen eindeutigen Pfad zwischen je zwei Knoten gibt.

(15 Punkte)

Aufgabe 4 (Algorithmus von Kruskal):



Bestimme mit Hilfe des Algorithmus von Kruskal einen minimalen aufspannenden Baum. Gib dazu die Kanten, in der Reihenfolge in der sie in den Baum aufgenommen werden, an und zeichne die gefundene Gesamtlösung. Kommen in einem Schritt mehrere Kanten in Frage, wähle die mit dem kleinsten Kantenindex.
Damit der Algorithmus von Kruskal eine Laufzeit von $O(m \log n)$ erreicht, kann man die in der Vorlesung vorgestellte Datenstruktur verwenden. Gib jeweils nach dem Einfügen einer Kante den Zustand der Datenstruktur an.
(Hinweis: Kommen beim Einfügen einer Kante in die Datenstruktur zwei Möglichkeiten in Frage, wähle die Kante, so dass sie vom Knoten mit dem kleineren Index zum Knoten mit dem größeren Index verläuft.)

(15 Punkte)