Abteilung Algorithmik
Institut für Betriebssysteme und Rechnerverbund **SoSe 14**
TU Braunschweig

Dr. Christiane Schmidt
Florian Maurer
Arne Schmidt

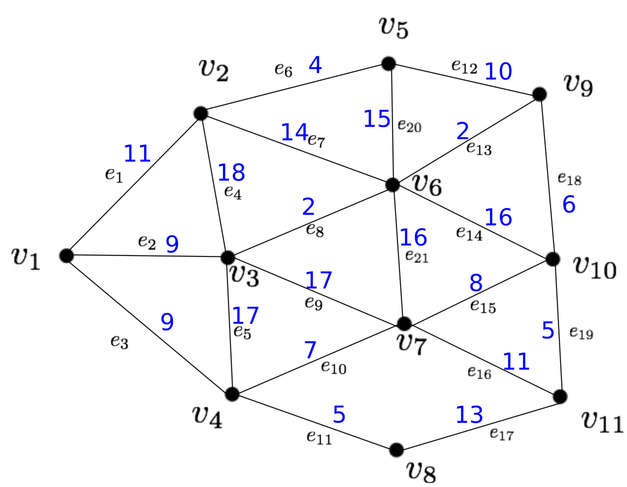
Netzwerkalgorithmen
Übung 2 vom 12.05.2014

Abgabe der Lösungen bis Mittwoch, den 28.05.14, bis 13:00 Uhr in der Abteilung *Algorithmik*.
Bitte die Blätter vorne deutlich mit eigenem Namen und Gruppennummer versehen!

Aufgabe 1 (Beweis Lemma 2.8):
Zeige: Gegeben sei ein Wald mit n Knoten, m Kanten und p Zusammenhangskomponenten. Dann gilt $n = m + p$.

(8 Punkte)

Aufgabe 2 (Algorithmus von Prim):



Bestimme mit Hilfe des Algorithmus von Prim einen minimalen aufspannenden Baum; beginne dabei mit dem Knoten v_1. (Hinweis: Kommen während einer Iteration mehrere Kanten in Frage, wähle die mit dem kleinsten Index.)

(11 Punkte)

Aufgabe 3 (Unabhängigkeitssysteme und Matroide):
Gegeben sei ein ungerichteter, zusammenhängender Graph G. Sei $E = E(G)$,
$\mathcal{I} = \{F \subseteq E : F \text{ ist Teilmenge eines Hamiltonkreises in } G\}$.

(a) Zeige, dass das Mengensystem $(E, \mathcal{I})$ ein Unabhängigkeitssystem ist.

(b) Prüfe, ob das Mengensystem $(E, \mathcal{I})$ ein Matroid ist.

Sei E_2 eine endliche Menge, k eine ganze, positive Zahl und $\mathcal{I}_2 = \{F \subseteq E_2 : |F| \leq k\}$.

(c) Prüfe, ob das Mengensystem $(E_2, \mathcal{I}_2)$ ein Matroid ist.

(5+5+5 Punkte)

Aufgabe 4 (Bäume): Seien (V, T_1) und (V, T_2) zwei Bäume auf derselben Knotenmenge V. Zeige: Für jede Kante $e \in T_1$ existiert eine Kante $f \in T_2$, so dass sowohl $(V, (T_1 \backslash \{e\}) \cup \{f\})$ als auch $(V, (T_2 \backslash \{f\}) \cup \{e\})$ Bäume sind.

(10 Punkte)

Aufgabe 5 (Minimal aufspannende Bäume):
Gegeben sei ein ungerichteter, zusammenhängender Graph $G = (V, E)$. Beweise oder widerlege die folgende Aussage:
Wenn es in G einen Kreis mit eindeutiger leichtester Kante e (= Kante mit geringstem Gewicht) gibt, dann ist e in jedem minimalen aufspannenden Baum enthalten.

(8 Punkte)

Aufgabe 6 (Bottleneck-Spannbaum):
In der Übung vom 12.5.2014 haben wir Bottleneck-Spannbäume betrachtet.
Gib einen Algorithmus an, der in Linearzeit für einen gegebenen Graph G und eine ganze Zahl b entscheidet, ob der Wert des Bottleneck Spannbaums höchstens b ist.

(8 Punkte)