Theoretische Informatik 1 / Algorithmentheorie

Wintersemester 2012/13

Prof. Dr. Georg Schnitger,
Dipl. Inf. Bert Besser,
Dipl. Inf. Matthias Poloczek

Arbeitsgruppe Theoretische Informatik, Institut für Informatik

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

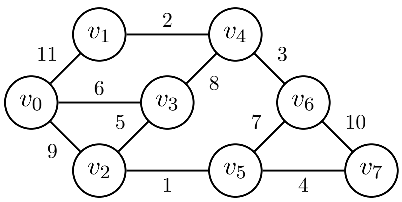
Blatt 4

Ausgabe: 15.11.2012
Abgabe: 22.11.2012 **vor** der Vorlesung

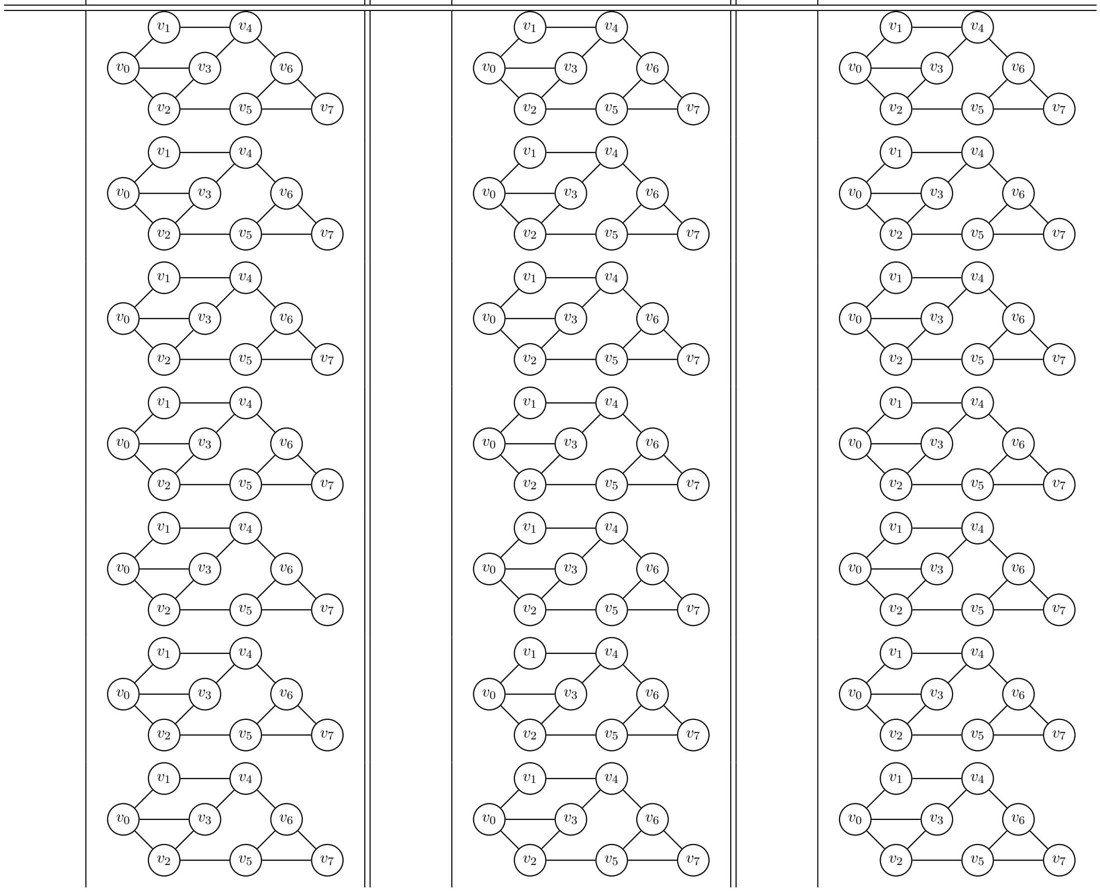
4.1. Aufgabe (2+2+2)

Minimale Spannbäume und kürzeste Wege

Wende für den Graphen rechts die Algorithmen von Kruskal, Prim und Dijkstra (je mit Startknoten v_0) an. Zeichne für Prim und Kruskal in jedem Schritt den bis dahin bestimmten Wald. Zeichne für Dijkstra in jedem Schritt den bis dahin bestimmten Baum kürzester Wege. Gib stets das Gewicht w der neuen Kante bzw. des entsprechenden S-Weges an.



w	Prim	w	Kruskal	w	Dijkstra



4.2. Aufgabe (8)

Achtung, Schwertransporter!

Ein Unternehmen für Schwertransporte beauftragt uns, für seine Lastwagen Routen mit einer möglichst geringen maximalen Steigung zu finden, da sich die Ladekapazität nach der maximalen Steigung entlang eines Weges richtet. Wir modellieren das Straßennetz als einen *gerichteten* Graphen $G = (V, E)$, in dem eine Kante mit der maximalen Steigung der entsprechenden Straße beschriftet ist. Die Zentrale der Firma liege im Knoten $s \in V$.

Wir müssen also nicht nur die kleinsten maximalen Steigungen für die Wege von s zu allen anderen Knoten bestimmen, sondern darüber hinaus auch die zugehörigen Wege berechnen. Entwirf hierzu einen möglichst effizienten Algorithmus und bestimme seine Laufzeit. Beweise unserem Auftraggeber, dass er sich auf den Algorithmus verlassen kann, d.h. dass der Algorithmus korrekt ist.

4.3. Aufgabe (4)

Schwieriges Problem fuer Greedy

In der Vorlesung haben wir einen Greedy-Algorithmus $\mathcal{G}$ für das Intervall-Scheduling-Problem kennengelernt, der eine größtmögliche Menge von kollisionsfreien Jobs wählt, indem er zuerst alle Jobs aufsteigend nach Terminierungszeit sortiert und dann gierig die Jobs mit kleinster Terminierungszeit übernimmt, wobei er jeweils alle kollidierenden Jobs verwirft.

Wir verändern die Aufgabenstellung, indem wir Jobs gewichten und nun eine schwerstmögliche Menge von kollisionsfreien Jobs fordern. Beweise, dass $\mathcal{G}$ in diesem Szenario versagt. Das Beispiel soll einen möglichst schlechten Approximationsfaktor aufweisen, d.h. das Verhältnis zwischen dem Gewicht der errechneten Menge und dem einer optimalen Menge soll möglichst klein sein.

4.4. Aufgabe (6)

Die Tourismusbranche

Einem Reisebegleitservice liegen für eine Menge von Reisegruppen Terminanfragen in Form von Ankunfts- und Abreisezeiten vor. Jeder Termin muss von genau einem Reisebegleiter betreut werden. Ein Begleiter kann mehrere Gruppen betreuen, solange die jeweiligen Termine nicht kollidieren.

Ein Begleiter erhält, unabhängig von der Anzahl der betreuten Gruppen, einen festen Tagessatz. Die Kosten für Mitarbeiter müssen kleinstmöglich sein, d.h. alle Termine müssen auf möglichst wenige Begleiter aufgeteilt werden. (Es kann davon ausgegangen werden, dass die Anzahl zur Verfügung stehender Begleiter in jedem Falle ausreichend ist.)

- Wie viele Begleiter müssen mindestens eingesetzt werden? Warum?
- Entwirf einen effizienten Algorithmus, der die Termine auf möglichst wenige Begleiter verteilt. Bestimme die Laufzeit und zeige die Korrektheit des Algorithmus.