

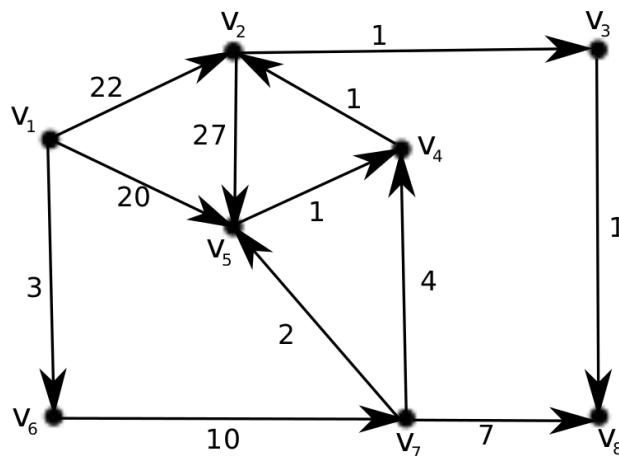
Prof. Dr. Sándor P. Fekete
Dr. Christiane Schmidt
Stephan Friedrichs

Netzwerkalgorithmen
Übung 3 vom 20.05.2013

Abgabe der Lösungen bis Mittwoch, den 05.06.12, bis 13:00 Uhr in der
Abteilung *Algorithmik*.

Bitte die Blätter vorne deutlich mit eigenem Namen und Gruppennummer versehen!

Aufgabe 1 (Algorithmus von Dijkstra):



Bestimme mit Hilfe des Algorithmus von Dijkstra einen kürzesten Weg von v_1 nach v_8 . (Hinweis: Kommen während einer Iteration mehrere Knoten in Frage, wähle den mit dem kleinsten Index.) Gib jeweils an, wenn sich Label und Vorgänger ändern.

(12 Punkte)

Aufgabe 2 (Bottleneck-Spannbaum):

In der Übung vom 13.5.2013 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.

(12 Punkte)

Aufgabe 3 (Dijkstra und MSTs):

Gegeben sei ein ungerichteter Graph G mit Kantengewichten $c : E(G) \mapsto \mathbb{R}$ (nicht paarweise verschieden).

Zeige oder widerlege: Der mit dem Dijkstra-Algorithmus berechnete Kürzeste-Wege-Baum ist auch ein minimaler aufspannender Baum.

(12 Punkte)

Aufgabe 4 (Dijkstra und konservative Gewichte):

Beweise oder widerlege: Falls c konservativ ist, findet der Algorithmus von Dijkstra einen kürzesten Weg von einem Knoten s zu allen anderen Knoten.

(12 Punkte)

Aufgabe 5 (Längste Wege und das Knapsack-Problem):

Betrachte n Objekte. Jedes dieser Objekte hat ein *Gewicht* a_j und einen *Wert* c_j , beides sind positive ganze Zahlen. Wir suchen nach einer Teilmenge dieser Objekte, so dass die Summe ihrer Gewichte eine Schranke b (die Größe des Knapsacks) nicht überschreitet und die Summe ihrer Werte maximiert wird (wir wollen also möglichst wertvolle Gegenstände einpacken).

Reduziere dieses Problem auf das Finden eines längsten Pfades in einem passenden Netzwerk.

Hinweis: Betrachte ein azyklisches Netzwerk mit einem Startknoten s , einem Endknoten t , und $b + 1$ Knoten für jedes Objekt.

(12 Punkte)