

Datenstrukturen und Algorithmen – Informatik I

11. Übung

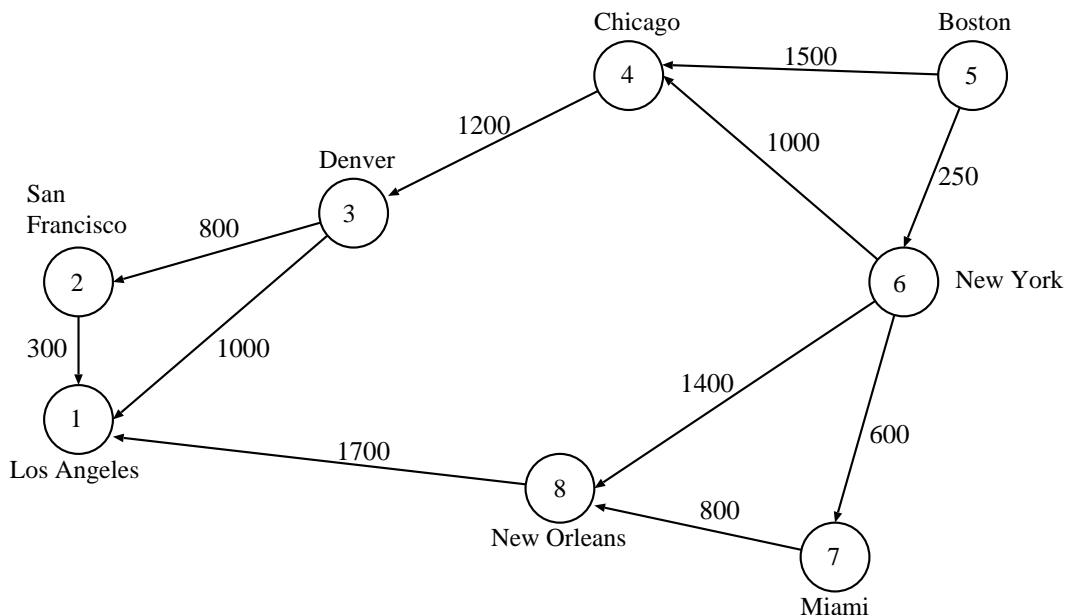
Abgabe der Lösungen: 19. Juli 2004

Hinweis: Die Implementierungsaufgaben sind in Java oder C/C++ zu lösen. Der Quellcode sowie ein dokumentierter Beispiellauf sind beim Betreuer der Kleingruppe abzugeben.

Aufgabe 48: Ausprobieren und modifizieren: Dijkstra-Algorithmus

5+1 Punkte

- a) Demonstrieren Sie die Arbeitsweise des Algorithmus von Dijkstra zur Bestimmung von Wegen mit minimalen Kosten anhand folgenden Beispiels:



1. Stellen Sie die Adjazenzmatrix mit den angegebenen Wegkosten auf.
2. Führen Sie mit der in Teil a) konstruierten Adjazenzmatrix den Dijkstra-Algorithmus durch. Der Startknoten sei "Boston".
 Notieren Sie nach jedem kompletten Rechenschritt (Update aller Matrixwerte) die Tabelle mit den aktuellen Kosten $D(S_k, v)$ für alle noch nicht abgearbeiteten Knoten v .
3. Geben Sie die besten Pfade in der Baumdarstellung an.
- b) Modifizieren Sie den Dijkstra-Algorithmus so, daß zu jedem Knoten nicht nur die Kosten des Pfades vom Startknoten aus, sondern auch die Anzahl der Wege mit minimalen Kosten vom Startknoten aus bestimmt wird. Geben Sie den resultierenden Algorithmus in Pseudocode an.

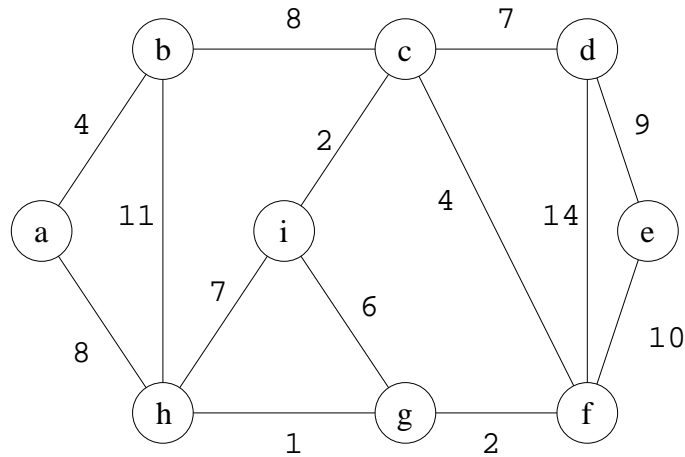
Aufgabe 49: Implementierung: Dijkstra-Algorithmus

5 Punkte

Implementieren Sie den Dijkstra-Algorithmus unter Verwendung von Adjazenzmatrizen in Java mit expliziter Konstruktion des besten Pfades. Testen Sie Ihren Algorithmus an dem oben angegebenen Graphen. Der Startknoten sei wieder "Boston".

Aufgabe 50: Prim-Algorithmus**5 Punkte**

Gegeben sei der folgende ungerichtete Graph:



Konstruieren Sie mit Hilfe des in der Vorlesung vorgestellten Prim-Algorithmus den minimalen Spannbaum.

Aufgabe 51: Wettbewerb: Dijkstra-Algorithmus**15+15* Punkte**

Implementieren Sie den Dijkstra-Algorithmus. Verwenden Sie Adjazenzlisten zur Repräsentierung der Kanten und eine (heap-basierte) erweiterten Prioritätswarteschlange zur Verwaltung der Randknoten, d.h. der Knoten, die von S_k aus erreicht worden sind und Kosten kleiner ∞ haben. In der Prioritätswarteschlange soll kein Knoten doppelt auftreten. Sie dürfen Ergebnisse aus vorherigen Aufgaben verwenden.

Auf der Webseite zur Vorlesung steht ein Rahmenwerk zu herunterladen bereit. Verändern Sie in diesem nur die Datei `DijkstraSolver.java`.

Zur Lösung gehört eine kurze Beschreibung der Arbeitsweise des Programms (ps, txt oder pdf).

Alle Teilnehmer einer Gruppe mit richtigen Lösungen erhalten 15 Punkte. Die Teilnehmer der Gruppe, die das schnellste Programm erstellt hat, und die Teilnehmer der Gruppe, die das kürzeste (gemessen an der Zahl der ; und () Programm erstellt hat, erhalten jeweils 15 Zusatzpunkte.

Abgabe für die Wettbewerbslösungen: Schicken Sie

- Ihren Java-Quellcode (inklusive Kommentare)
- eine kurze Beschreibung des Programms

per e-mail an `deselaers@i6.informatik.rwth-aachen.de`.

Abgabeschluß ist **Montag, der 19. Juli 2004 um 12h**.

* 15 Zusatzpunkte im Rahmen des Wettbewerbs. Die anderen 15 Punkte zählen zu den normalen Übungspunkten.

Hinweis: Diese Übung ist eine *Auswahlübung*, d.h. von den 31 Punkten dieser Übung werden nur 25 bei der Berechnung der zu erreichenden Gesamtpunktzahl berücksichtigt. Wenn eine Gruppe mehr als 25 Punkte erreicht, so sind dies Zusatzpunkte, die normal gewertet werden.