

**Algorithmen und Datenstrukturen**  
SS 2017

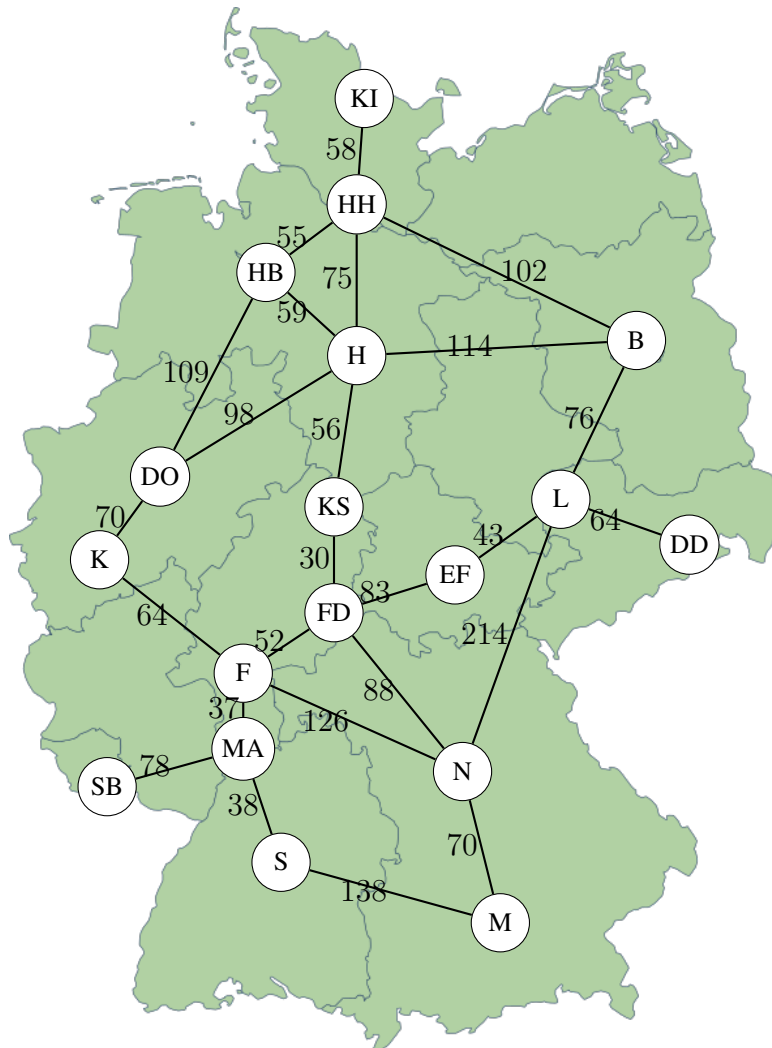
**Übungsblatt 10: Kürzeste Wege**

Besprechung: 10.07-14.07.2017

Ende der Abgabefrist: 10.07.2017 - 12:00 Uhr.

**Aufgabe 10-1**     *Kürzeste Wege*

**10 Punkte**



Die obige Karte zeigt das Streckennetz einiger ICE-Direktverbindungen zwischen größeren deutschen Städten als ungerichteten Graphen. Als Kantengewichte sind die geschätzten Zeiten in Minuten angegeben, die ein Zug zwischen beiden Bahnhöfen braucht.

- (a) Wenden Sie analog zur Vorlesung den Dijkstra-Algorithmus auf das Streckennetz an. Finden Sie so den Subgraphen, der die kürzesten Wege von München zu allen anderen Städten enthält. Geben Sie nach jeweils drei betrachteten Knoten Zwischenschritte an (insgesamt also 6 Graphen).
- (b) Implementieren Sie den Floyd-Algorithmus in Java und bestimmen Sie alle kürzesten Wegpaare zwischen jeweils zwei Städten aus. Geben Sie nur die Ergebnismatrix an, keine Implementierung. Verwenden Sie die gegebene Reihenfolge der Zeilen/Spalten. Die Adjazenzmatrix können Sie auf der Webseite herunterladen. Ein Ergebnis könnte in etwa wie folgt aussehen:

	K	DO	HB	HH	KI	H	B	KS	L	FD	EF	DD	F	N	MA	SB	S	M
K																		
DO																		
HB																		
HH																		
KI																		
H																		
B																		
KS																		
L																		
FD																		
EF																		
DD																		
F																		
N																		
MA																		
SB																		
S																		
M																		

- (c) In den Vorlesungsfolien operieren diese Algorithmen auf gerichteten Graphen. Führt dies zu einem Problem? Warum (nicht)?
- (d) Nun wird das Transportnetzwerk durch ein paar Flüge in jeweils eine Richtung ergänzt:  
 München → Hamburg(75)  
 Hamburg → Frankfurt (70)  
 Frankfurt → Berlin (70)  
 Berlin → München (65)  
 Geben Sie die sich nun ergebende Distanzmatrix aus. Wie unterscheiden sich diese und die Matrix aus Aufgabe (b)?
- (e) \* (**Erst auf Blatt 11 zur Bewertung enthalten**) Wenden Sie den Prim-Algorithmus auf das Streckennetz ohne Flüge an. Geben Sie den resultierenden minimalen Spannbaum an. Beginnen Sie mit München als Startknoten. Nummerieren Sie die Kanten in der Reihenfolge, in der sie hinzugefügt werden.
- Hinweis: Die zugehörige Adjazenzmatrix können Sie der Datei `distance.java` entnehmen. Wartezeiten und Verspätungen sollen nicht berücksichtigt werden. Gehen Sie davon aus, dass ein Reisender ohne Zeitverlust Züge wechseln kann und das jederzeit ein exklusiver ICE auf ihn wartet. Auf einem Extrablatt finden Sie eine Vorlage des Graphen, die Sie für Aufgabe (a) und (e) benutzen können.*

