

## Miniprojekt 1

Abgabe bis Freitag, 12.1.2018, 9.45 Uhr

### Graphen

Die untenstehende ‘Landkarte’ lässt sich als gewichteter Graph interpretieren, bei dem die Städte den Ecken, die Verbindungswege den Kanten und die Weglängen den Kantengewichten entsprechen. Die kürzeste Weglänge zwischen zwei beliebigen Städten lässt sich mit dem Min-Plus-Matrixmultiplikations-Algorithmus bestimmen, der auf folgende Weise implementiert werden kann:

Gegeben sei ein Graph  $G = (E, K)$  und eine Adjazenzmatrix  $A$  mit Gewichten  $a_{i,j}$ , wobei die Indizes  $i$  und  $j$  über die Eckenmenge  $E$  laufen.

- (1) Bestimmen sie die Kostenmatrix  $C$  durch

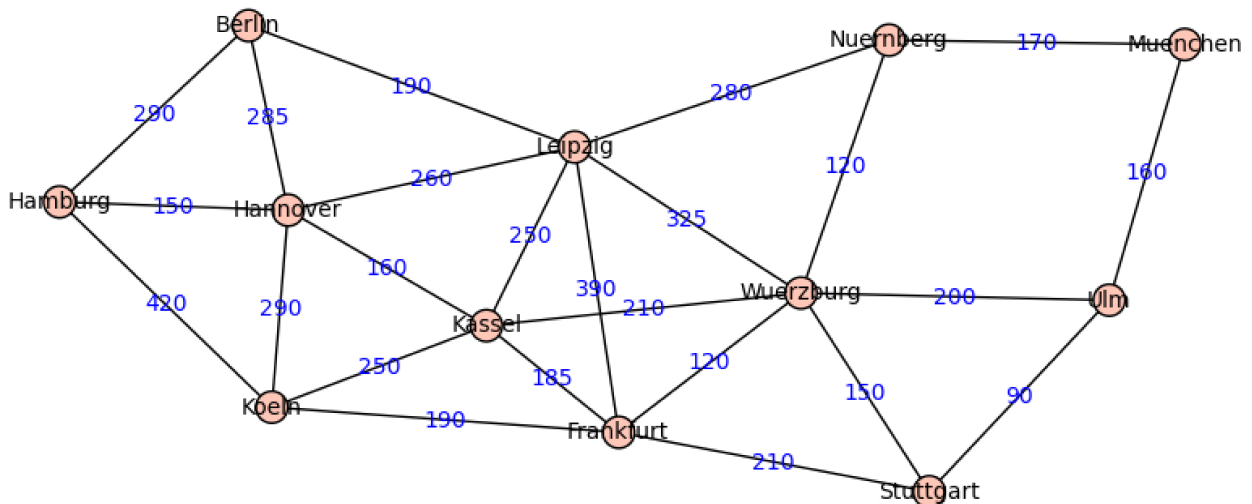
$$c_{i,j} = \begin{cases} 0 & \text{falls } i = j \\ a_{i,j} & \text{falls } (i,j) \in K \\ \infty & \text{sonst.} \end{cases}$$

- (2) Definieren sie sich für zwei  $n \times n$  Matrizen  $A$  und  $B$  eine Matrizenoperation  $C = A \oplus B$  durch

$$c_{i,k} = \min\{a_{i,j} + b_{j,k} \mid j \in \{1, \dots, n\}\},$$

wobei gelten soll  $c + \infty = \infty + c = \infty$ . Außerdem gilt die Schreibweise  $C \oplus C = C^{[2]}$ .

- (3) Die Matrix  $C^{[p]}$  gibt die Länge der kürzesten Pfade mit maximal  $p$  Kanten an, wobei  $c_{i,j}^{[p]}$  dem kürzesten Pfad von Ecke  $i$  zu  $j$  entspricht.



### Aufgaben:

- (a) Schreiben sie eine Funktion  $Kostenmatrix(A)$  mit der Adjazenzmatrix  $A$  als Input, welche die zugehörige Kostenmatrix ausgibt.

**Hinweis:** Anstatt  $\infty$  können sie auch den Wert  $10^{10}$  einsetzen, da in Sage ein Matrixelement nicht den Wert  $\infty$  annehmen kann.

- (b) Schreiben sie eine Funktion  $Matrizenoperation(A,B)$  mit zwei  $n \times n$  Matrizen als Input, welche die Matrizenoperation aus dem in Teil (2) erklärten Algorithmus umsetzt.

**Hinweis:** Der Befehl  $A.nrows()$  gibt die Zeilenanzahl einer Matrix  $A$  zurück.

- (c) Die Entfernungsmatrix  $D$  eines gewichteten Graphen sei definiert durch

$$d_{i,j} = \begin{cases} 0, & \text{falls } i = j \\ \text{Länge des kürzesten Weges von Knoten } i \text{ zu } j \\ \infty, & \text{falls es keinen Weg gibt.} \end{cases}$$

Schreiben sie eine Funktion  $kurzerWeg(A)$  mit der Gewichtsmatrix  $A$  eines Graphen  $G$  als Input, welche die Entfernungsmatrix  $D$  des Graphen  $G$  zurückgibt.

Programmieren sie die Funktion effizient, indem sie so wenige Matrizenoperationen wie möglich verwenden.

- (d) Plotten sie den Graphen auf der vorherigen Seite.

**Hinweis:** Die folgenden Befehle sind hilfreich

$g = \text{Graph}(A, \text{weighted} = \text{True})$ , wobei  $A$  eine Adjazenzmatrix ist.

$g.\text{relabel}(\text{Liste})$  benennt die Knoten um, durch Namen aus einer Liste.

$g.\text{plot}(\text{edge\_labels} = \text{True})$  fügt die Kantengewichte dem Plot hinzu.

Die genaue Position der Knoten kann variieren und muss nicht beachtet werden.

- (e) Bestimmen sie mit ihrer Funktion aus (c) den kürzesten Weg von Hamburg nach München und von Berlin nach Stuttgart, indem sie die Matrix  $D$  ausgeben und die Entfernungen ablesen.

Finden und markieren sie diese beiden kürzesten Wege in ihrem Graphen farbig.

- (f) Schreiben sie ein LaTeX-Dokument mit Titelseite und einer Dokumentation ihrer Ergebnisse, sowie dem vollständigen Lösungsweg der einzelnen Teilaufgaben.