Algorithmen und Datenstrukturen Übung 6

1. Berechnung der kürzesten Pfade mithilfe einer Vorgängermatrix:

   gegeben: eine Menge von Knoten V = 1,2, ..., n und ein gerichteter Graph G = (V,E), ohne negative Zyklen

   Idee:
   Zusätzlich zu der Distanzmatrix nutzt man noch eine Vorgängermatrix, in welcher die nötigen Informationen für die Generierung der kürzesten Wege gespeichert werden.

   V[i,j] = 0, falls j von i aus nicht erreichbar ist
   V[i,j] = i, falls i = j ist
   V[i,j] = k, falls k der Vorgänger von j auf einem kürzesten Weg von i nach j ist

   Der kürzeste Weg wird nun, wie folgt bestimmt:
   Sei k_1 = V[i,j]. Mit k_1 als den Vorgänger von j auf dem kürzesten Weg zwischen i und j.
   Sei k_2 = V[i, k_1, k_3 = V[i, k_2,... , bis ein k_x = i wird.
   Dann ist [i, k_{x-1},... , k_2, k_1, j] ein kürzester Weg der Länge D[i,j] von i nach j.

   Pseudocode:

   // Initialisierung der Distanzmatrix und der Vorgängermatrix

   forall Knotenpaare(i,j) ∈ (N x N) do
   D[i,j] = ∞
   V[i,j] = 0
   end forall

   forall Knoten i ∈ N do
   D[i,i] = 0
   V[i,i] = i
   end forall

   forall Kanten(i,j) ∈ A do *// A = Adjazenzmatrix*
   D[i,j] = c(i,j)
   V[i,j] = i
   end forall

   for k = 1 to n do *// für alle Knoten k von 1 bis n*
   forall Knotenpaare (i,j) ∈ (N x N) do
   if D[i,k] + D[k,j] < D[i,j] then

   // in den folgenden Zeilen wird, wenn möglich der kürzeste Weg von i nach j über den Knoten k verbessert

```
            D[i,j] = D[i,k] + D[k,j]
            V[i,j] = V[k,j]

// hier wird sich der Vorgänger von j gemerkt

          end if
        end forall
end for
```

Der obige Algorithmus ist 'optisch' von dem in der Vorlesung präsentiertem leicht abgewandelt, allerdings inhaltlich gleich, was die Berechnung der Distanzmatrix betrifft. Somit wissen wir, dass die Länge der Kürzesten-Pfad-Distanz zwischen allen Knoten korrekt berechnet wird. Also auch wird auch der kürzeste Weg korrekt berechnet, nur noch nicht abgespeichert.
Der Algorithmus tut nun nichts weiter, als sich den Vorgängerknoten von j zu merken. Um den kürzesten Weg zu bestimmen muss man diesen nur noch mit oben beschrieben Schema auslesen, indem man bei j anfängt und in der Vorgängeratrix jeweils den Vorgänger sucht, bis man bei i angelangt ist.

vgl. Buch: 'Optimierungssysteme: Modelle, Verfahren, Software, Anwendungen' von Leena Suhl,Taïeb Mellouli