I.4 Warshall - Algorithmus

Der Dijkstra - Algorithmus bietet eine relativ schnelle Möglichkeit den minimalen Weg zwischen zwei Knoten in einem Graphen zu bestimmen. Bei anderer Abbruchbedingung erhält man auch den minimalen Weg von einem Knoten zu allen anderen Knoten.

Zum Abschluss unserer Wegesuche wollen wir nun ein ähnliches Problem betrachten.

Wie findet man die minimalen Wege von allen Knoten zu allen anderen Knoten ?

Diese Problem lässt sich natürlich ebenfalls mit dem Dijkstra - Verfahren lösen. Dazu wendet man den Dijkstra - Algorithmus auf alle Knoten als Startknoten an. Es gibt jedoch noch andere Verfahren, die das Problem direkt angehen. Eines dieser Verfahren ist der Warshall - Algorithmus.

Betrachten wir dazu zuerst ein etwas einfacheres Beispiel:
Für die Städte München, Paris, London, Amsterdam und Boston liegen die folgenden billigsten Direktverbindungen vor (einfache Wegstrecke). Für die umgekehrten Wegstrecken gelten die gleichen Preise.

München	–	Paris	:	150.- DM
London	–	Amsterdam	:	50.- DM
Paris	–	Boston	:	300.-DM
Paris	–	Amsterdam	:	100.- DM
London	–	Boston	:	275.- DM
München	–	London	:	99.- DM
Amsterdam	–	Boston	:	290.- DM

	M	P	L	A	B
M	0	150	99	/	/
P	150	0	/	100	300
L	99	/	0	50	275
A	/	100	50	0	290
B	/	300	275	290	0

In dieser Kostenmatrix sind nur die direkten Verbindungen eingetragen. Die Idee des Warshall - Verfahrens beruht nun darauf, nacheinander sämtliche Wege über irgend welche Zwischenknoten zu betrachten. Beginnen wir in der Matrix, so betrachten wir als erstes die Flüge über München. Diese Flüge von und nach München sind in der ersten Zeile bzw. in der ersten Spalte eingetragen.

Beschreibung des Warshall - Verfahrens.

> Die Spalte und die Zeile des betrachteten Zwischenknotens wird markiert. In der neuen Matrix wird nun die Diagonale mit 0 belegt und danach werden die markierten Elemente eingetragen. Anschließend betrachtet man alle freigebliebenen Felder. Ist der Eintrag in dem Feld größer als die Summe der markierten Felder in der gleichen Spalte und. in der gleichen Zeile, so wird der Eintrag durch diese Summe ersetzt. Sonst bleibt er bestehen. Dies wird wiederholt bis man alle Zwischenknoten betrachtet hat.

Dieses Verfahren soll nun an der obigen Beispielmatrix durchgeführt werden. Man beginnt mit den Wegen über München. Dazu markieren man die erste Spalte und die erste Zeile und überträgt sie, wie auch die Diagonalelemente in die neue Matrix. Nun werden die freigebliebenen Einträge nach dem obigen Schema bearbeitet.

Flüge über München

	M	P	L	A	B
M	0	150	99	/	/
P	150	0	/	100	300
L	99	/	0	50	275
A	/	100	50	0	290
B	/	300	275	290	0

	M	P	L	A	B
M	0	150	99	/	/
P	150	0	249	100	300
L	99	249	0	50	275
A	/	100	50	0	290
B	/	300	275	290	0

Damit ergibt sich die rechtstehende Matrix. Als nächstes betrachtet man nun die Flüge über Paris. Dabei werden wieder die entsprechenden Zeilen und Spalten in der neuen Matrix markiert und daraus wieder eine neue Matrix erstellt.

Flüge über Paris

	M	P	L	A	B
M	0	150	99	/	/
P	150	0	249	100	300
L	99	249	0	50	275
A	/	100	50	0	290
B	/	300	275	290	0

	M	P	L	A	B
M	0	150	99	250	450
P	150	0	249	100	300
L	99	249	0	50	275
A	250	100	50	0	290
B	450	300	275	290	0

Diese Verfahren muss nun fortgesetzt werden bis alle Zwischenknoten bearbeitet werden.

Nach Beendigung des Verfahrens erhält man nebenstehende Matrix. Aus ihr lassen sich nun die günstigsten Flugpreise (oder kürzesten Wegstrecken ...) zwischen zwei Orten bestimmen.
Allerdings lässt sich nicht ermitteln, wo man umsteigen muss. Der Weg wird damit nicht angegeben.

	M	P	L	A	B
M	0	150	99	149	374
P	150	0	150	100	300
L	99	150	0	50	275
A	149	100	50	0	290
B	374	300	275	290	0

Bestimmung des Weges beim Warshall - Verfahren

Um den Weg zu ermitteln muss neben der Matrix für die minimalen Strecken eine zweite Matrix geführt werden, die Wegematrix. In ihr werden alle direkten Wege mit einem Stern markiert. Findet man nun bei einem Schritt des Warshall - Verfahrens einen neuen Weg, so wird in dem entsprechenden Feld der Wegematrix der aktuelle Zwischenknoten eingetragen.
Betrachten wir dies an dem folgenden Beispiel:

Bewertete Adjazenzmatrix

	A	B	C	D	E
A	0	10	6	11	10
B	12	0	/	/	/
C	/	3	0	/	/
D	/	4	/	0	/
E	/	/	/	2	0

Wegematrix

	A	B	C	D	E
A	*	*	*	*	*
B	*	*	/	/	/
C	/	*	*	/	/
D	/	*	/	*	/
E	/	/	/	*	*

Als nächstes betrachten wir nun die Wege über A und erhalten damit:

Wege über A

	A	B	C	D	E
A	0	10	6	11	10
B	12	0	18	23	22
C	/	3	0	/	/
D	/	4	/	0	/
E	/	/	/	2	0

	A	B	C	D	E
A	*	*	*	*	*
B	*	*	A	A	A
C	/	*	*	/	/
D	/	*	/	*	/
E	/	/	/	*	*

Am Ende erhält man nun:

	A	B	C	D	E
A	0	9	6	11	10
B	12	0	18	23	22
C	15	3	0	26	25
D	16	4	22	0	26
E	18	6	24	2	0

	A	B	C	D	E
A	*	C	*	*	*
B	*	*	A	A	A
C	B	*	*	B	B
D	B	*	B	*	B
E	D	D	D	*	*

In der Wegematrix ist nun der Weg nicht direkt auslesbar, wie folgende Beispiele zeigen.

Weg von D nach B:	D - B	(Hier gibt es einen direketen Weg)
Weg von C nach A:	C - A	(Hier gibt es einen Weg über B)
	C - B - A	(Sowohl von C nach B, wie von B nach A gibt es einen direkten Weg)
Weg von E nach C:	E - C	(Hier gibt es einen Weg über D)
	E - D - C	(Von D nach C geht der Weg über B)
	E - D - B - C	(Von B nach C geht der Weg über A)
	E - D - B - A - C	(Nun sind alle Wege direkte Verbindungen)

Die Wege über Zwischenknoten müssen solange bearbeitet werden, bis alle Teilstrecken direkte Verbindungen darstellen.

Realisierung des Warshall Verfahrens

Nachdem nun die Grundlagen des Warshall - Verfahrens geklärt sind, soll es nun auf dem Rechner realisiert werden. Dazu muss zuerst die Datenstruktur für die beiden Matrizen festgelegt werden. Die bewertete Adjazenzmatrix ist die alte Entfernungs - oder Zeitmatrix. Aus ihr wird die Warshallmatrix entwickelt. Die Wegematrix ist vom gleichen Typ. Die direkten Wege wurden bisher durch das Zeichen „*“ gekennzeichnet. Dies ist hier jedoch nicht möglich, da in die Matrix nur Zahlen eingetragen werden können. Daher soll die Zahl -1 eine direkte Verbindung kennzeichnen. (0 ist nicht möglich, da damit der erste Ort gekennzeichnet wird). Nicht vorhandene Wege werden dann durch eine große Zahl (10000) gekennzeichnet.

Datenstruktur des Warshall - Verfahrens

```
VAR
  Entfernung_Warshallmatrix    : tMatrix;
  Entfernung_Wegematrix        : tMatrix;
                               (* Direkte Wege werden durch -1
                               gekennzeichnet, 10000 bedeutet
                               keine Verbindung *)
  Zeit_Warshallmatrix          : tMatrix;
  Zeit_Wegematrix              : tMatrix;
```

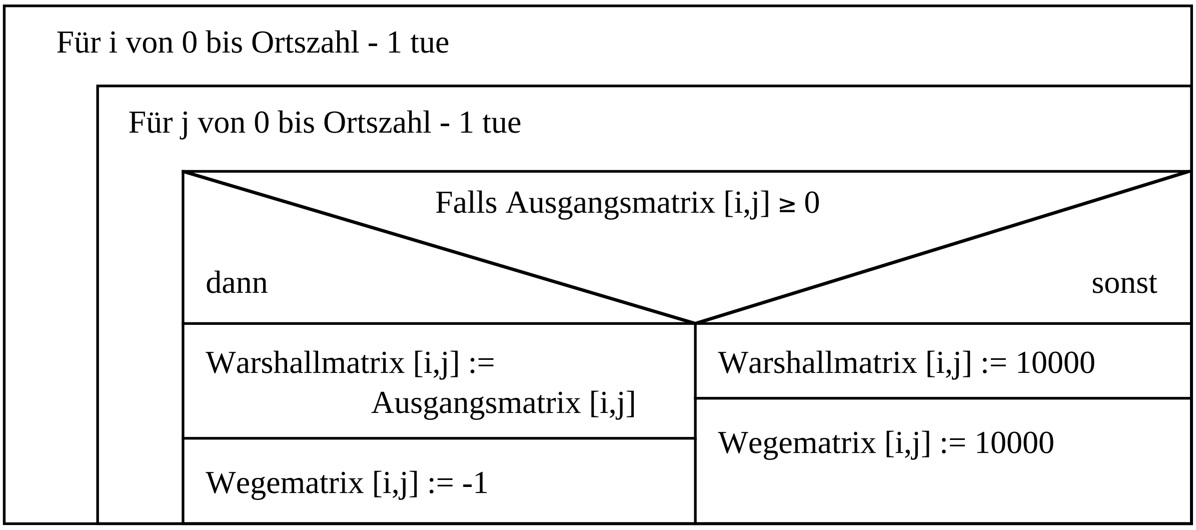
Programmstruktur:
Im Gegensatz zu den bisherigen Programmen ist es günstiger die Matrizen (Warshall- und Wegematrix) am Anfang auszurechnen, da die Berechnung nur einmal durchgeführt werden muss. Erst danach erfolgt die Eingabe von Start- und Zielort. Damit ergibt sich folgender Aufbau..Nach dem Laden der Daten von der Festplatte werden die beiden Matrizen initialisiert und berechnet.

Datei. Laden (Entfernungsmatrix, Zeitmatrix, Orte, Ortszahl)
Berechne.Initialisieren (Entfernungsmatrix, Ortszahl, Entfernung_Warshallmatrix, Entfernung_Wegematrix)
Berechne.Initialisieren (Zeitmatrix, Ortszahl, Zeit_Warshallmatrix, Zeit_Wegematrix)
Berechne.Warshall (Entfernung_Warshallmatrix, Entfernung_Wegematrix, Ortszahl)
Berechne.Warshall (Zeit_Warshallmatrix, Zeit_Wegematrix, Ortszahl)
Menue
Datei.Sichern (Entfernungsmatrix, Zeitmatrix, Orte, Ortszahl)

Betrachten wir nun die einzelnen Prozeduren. Beginnen wir mit dem Initialisieren.

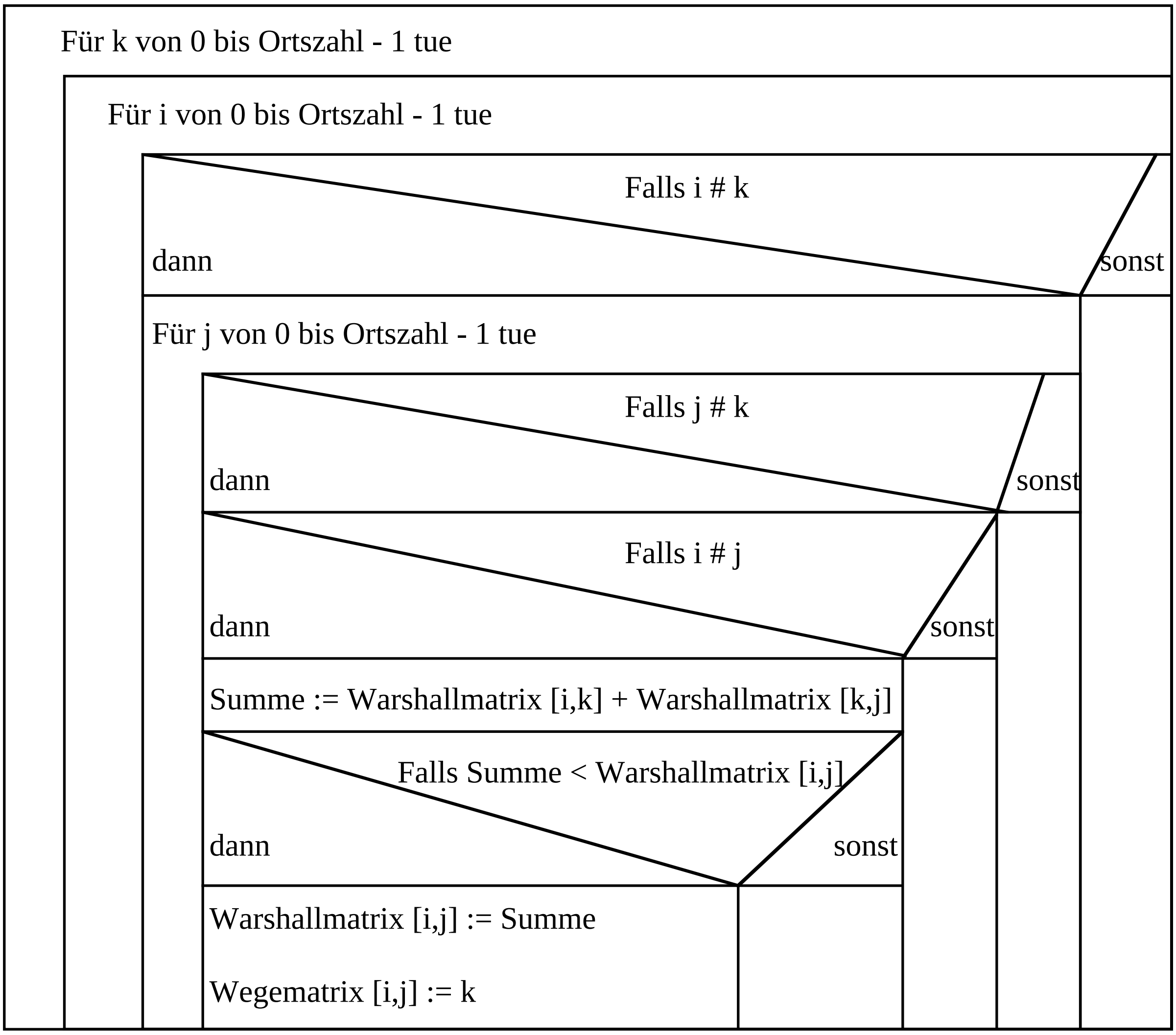
PROCEDURE Initialisieren (Ausgangsmatrix : tMatrix; Ortszahl : INTEGER;
VAR Warshallmatrix , Wegematrix: tMatrix);

Immer wenn in der Ausgangsmatrix ein Weg existiert (≥ 0) dann wird in die Warshallmatrix der Wert aus der Ausgangsmatrix übernommen und in die Wegematrix die Zahl -1 eingetragen. Sonst trägt man in beide Matrizen die Zahl 10000 ein.



Nachdem die Matrizen initialisiert sind, kann nun das Warshall - Verfahren durchgeführt werden. Dies geschieht in der Prozedur Warshall.

PROCEDURE Warshall (VAR Warshallmatrix, Wegematrix : tMatrix; Ortszahl : INTEGER);



Nun fehlt nur noch die Ausgabe der Ergebnisse. Die Ausgabe der Entfernung, bzw. der Fahrzeit ist sehr einfach. Sie steht bereits in der Warshallmatrix als Eintrag und muss nur ausgelesen werden.
Etwas aufwendiger ist die Ausgabe des Weges. Er muss aus der Wegematrix ermittelt werden. Dort stehen die Zwischenknoten und man betrachtet dann die Wege zu den Zwischenknoten. Dies wird solange verfeinert, bis man bei den direkten Wegen angekommen ist.

PROCEDURE Weg_ausgeben (Start, Ziel : INTEGER; Matrix : tMatrix
Orte : tOrte);

Falls Matrix [Start, Ziel] = -1	
dann	sonst
Ausgabe (Orte [Start])	Zwischenknoten := Matrix [Start, Ziel]
	Weg_ausgeben (Start, Zwischenknoten, Matrix, Orte)
	Weg_ausgeben (Zwischenknoten, Ziel, Matrix, Orte)

Diese Prozedur gibt alle Orte des Weges aus, nur der Zielort fehlt. Dieser Ort muss ganz am Schluss zusätzlich auf den Bildschirm geschrieben werden.
Dazu muss in die Prozedur SchnellerWeg bzw. KurzerWeg der Ausgabeteil eingefügt werden. In KurzerWeg wird eingefügt:

Weg_ausgeben (Startkennung, Zielkennung, Entfernung_Wegematrix, Orte)
Ausgabe (Orte [Zielkennung]);
Ausgabe („Die Wegstrecke beträgt : “)
Ausgabe (Entfernung_Warshallmatrix [Startkennung, Zielkennung])
Ausgabe („km“)

In der Prozedur SchnellerWeg müssen jeweils die Zeit_Warshallmatrix und Zeit_Wegematrix verwendet sowie der Ausgabetext angepasst werden.

Bewertung des Verfahrens

Um auch hier die Güte des Algorithmus zu beurteilen, wollen wir versuchen den Aufwand in Abhängigkeit der Knotenzahl zu bestimmen. Betrachten wir dabei wieder den Fall von n Orten.

Beim Dijkstra - Verfahren mussten wir n^2 Orte betrachten, um den Weg von einem Ort zu allen anderen Orten zu ermitteln. Wie wir oben gesehen haben, kann mit dem Dijkstra - Verfahren auch der Weg von allen Orten zu allen Orten bestimmt werden. Dazu muss das Dijkstra - Verfahren mit allen Orten als Startort durchgeführt werden. Damit ergeben sich $n \cdot n^2 = n^3$ zu betrachtende Orte.
Versucht man nun das Problem mit dem Warshall - Verfahren zu lösen, so muss man das Struktogramm der Prozedur Warshall betrachten. Dort treten drei ineinander geschachtelte Wiederholungsanweisungen auf, wobei die Anzahl der Wiederholungen jeweils n beträgt. Damit werden insgesamt $n \cdot n \cdot n = n^3$ Orte betrachtet. Damit ist der Aufwand beim Dijkstra- und beim Warshall - Verfahren gleich. Neben dem zeitlichen Aufwand ist auch noch der Speicheraufwand zu sehen. Hier hatten wir beim Dijkstra-Verfahren einen Vektor bestehend aus Records und beim Warshall-Verfahren zwei Felder. Hier dürften die Felder etwas günstiger sein, da der Record 5 Einträge hatte.