

Eulenfest - Glühwein statt Impfung

Di. 15.12. im Infobau - ab 19 Uhr
Musik - Glühwein - Bier - Waffeln - Grill - Cocktails
Mithelfen: www.fsmi.uni-karlsruhe.de/helfen



Grundbegriffe der Informatik

Einheit 12: Erste Algorithmen in Graphen

Thomas Worsch

Karlsruher Institut für Technologie, Fakultät für Informatik

Wintersemester 2009/2010

Repräsentation von Graphen im Rechner

Berechnung der 2-Erreichbarkeitsrelation und Rechnen mit Matrizen

- 2-Erreichbarkeit an einem Beispiel

- Matrixmultiplikation

- Matrixaddition

Einfache Berechnung der Erreichbarkeitsrelation

- Potenzen der Adjazenzmatrix

- Erste Möglichkeit für die Berechnung der Wegematrix

- Zählen durchzuführender arithmetischer Operationen

- Weitere Möglichkeiten für die Berechnung der Wegematrix

Algorithmus von Warshall

Repräsentation von Graphen im Rechner

Berechnung der 2-Erreichbarkeitsrelation und Rechnen mit Matrizen

- 2-Erreichbarkeit an einem Beispiel

- Matrixmultiplikation

- Matrixaddition

Einfache Berechnung der Erreichbarkeitsrelation

- Potenzen der Adjazenzmatrix

- Erste Möglichkeit für die Berechnung der Wegematrix

- Zählen durchzuführender arithmetischer Operationen

- Weitere Möglichkeiten für die Berechnung der Wegematrix

Algorithmus von Warshall

```
class Vertex {  
    String name;           // oder was auch immer  
}  
  
class Edge {  
    Vertex start;  
    Vertex end;  
}  
  
class Graph {  
    Vertex[] vertices;  
    Edge[] edges;  
}
```

```
class Vertex {  
    String name;           // oder was auch immer  
}
```

```
class Edge {  
    Vertex start;  
    Vertex end;  
}
```

```
class Graph {  
    Vertex[] vertices;  
    Edge[] edges;  
}
```

```
class Vertex {  
    String name;           // oder was auch immer  
}  
  
class Edge {  
    Vertex start;  
    Vertex end;  
}  
  
class Graph {  
    Vertex[] vertices;  
    Edge[] edges;  
}
```

```
class Vertex {  
    int id;  
}  
  
class Edge {  
    Vertex start;  
    Vertex end;  
}  
  
class Graph {  
    Vertex[] vertices;  
    Edge[] edges;  
}
```



```
class Vertex {  
    int id;  
    Vertex[] neighbors;    // Feldlänge = Knotengrad  
}  
  
class Edge {  
    Vertex start;  
    Vertex end;  
}  
  
class Graph {  
    Vertex[] vertices;  
    Edge[] edges;  
}
```

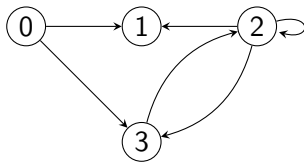
```
class Vertex {  
    int id;  
    Edge[] incoming;  
    Edge[] outgoing;  
}  
  
class Edge {  
    Vertex start;  
    Vertex end;  
}  
  
class Graph {  
    Vertex[] vertices;  
    Edge[] edges;  
}
```

```
class Vertex {  
    int id;  
    boolean[] is_connected_to;    // Feldlänge = |V|  
}
```

```
class Graph {  
    Vertex[] vertices;  
}
```

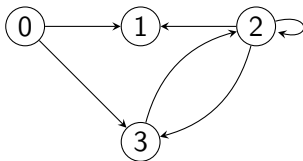
- ▶ Knoten: Objekte u, v der Klasse Vertex

- ▶ $u.is_connected_to[v.id] = \begin{cases} \text{true} & \text{falls } (u, v) \in E \\ \text{false} & \text{falls } (u, v) \notin E \end{cases}$



Objekt u , das Knoten 0 repräsentiert:

$u.id$	$u.is_connected_to$			
0	false	true	false	true
	0	1	2	3



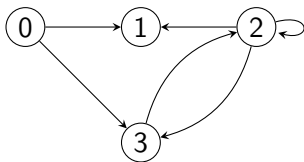
Objekte für alle Knoten untereinander:

<i>u.id</i>	<i>u.is_connected_to</i>			
0	false	true	false	true
1	false	false	false	false
2	false	true	true	true
3	false	false	true	false
	0	1	2	3

- Adjazenzmatrix eines gerichteten Graphen $G = (V, E)$ mit $|V| = n$ ist eine $n \times n$ -Matrix A mit der Eigenschaft:

$$A_{ij} = \begin{cases} 1 & \text{falls } (i, j) \in E \\ 0 & \text{falls } (i, j) \notin E \end{cases}$$

- Beispiel:



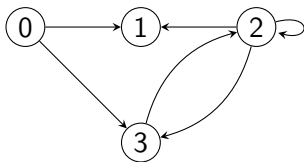
$$\begin{array}{c} \begin{matrix} & 0 & 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

- Adjazenzmatrix eines ungerichteten Graphen $U = (V, E)$ ist die Adjazenzmatrix von $G = (V, E_g)$

- Adjazenzmatrix eines gerichteten Graphen $G = (V, E)$ mit $|V| = n$ ist eine $n \times n$ -Matrix A mit der Eigenschaft:

$$A_{ij} = \begin{cases} 1 & \text{falls } (i, j) \in E \\ 0 & \text{falls } (i, j) \notin E \end{cases}$$

- Beispiel:



$$\begin{array}{c} \begin{matrix} & 0 & 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

- Adjazenzmatrix eines ungerichteten Graphen $U = (V, E)$ ist die Adjazenzmatrix von $G = (V, E_g)$

- ▶ endliche Menge M mit n Elementen
- ▶ binäre Relation $R \subseteq M \times M$
- ▶ repräsentiert durch $n \times n$ -Matrix $A(R)$:

$$(A(R))_{ij} = \begin{cases} 1 & \text{falls } (i,j) \in R \quad \text{d. h. also } iRj \\ 0 & \text{falls } (i,j) \notin R \quad \text{d. h. also } \neg(iRj) \end{cases}$$

- ▶ zu verschiedenen Relationen gehören verschiedene Matrizen und umgekehrt

- ▶ Erreichbarkeitsrelation E^* als Matrix repräsentierbar
- ▶ die sogenannte *Wegematrix* W des Graphen:

$$\begin{aligned} W_{ij} &= \begin{cases} 1 & \text{falls } (i,j) \in E^* \\ 0 & \text{falls } (i,j) \notin E^* \end{cases} \\ &= \begin{cases} 1 & \text{falls es in } G \text{ einen Pfad von } i \text{ nach } j \text{ gibt} \\ 0 & \text{falls es in } G \text{ keinen Pfad von } i \text{ nach } j \text{ gibt} \end{cases} \end{aligned}$$

- ▶ algorithmisches Problem:
 - ▶ gegebene Problemistanz: Adjazenzmatrix eines Graphen
 - ▶ gesucht: zugehörige Wegematrix des Graphen

Das sollten Sie mitnehmen:

- ▶ Repräsentation von Relationen als Matrizen
- ▶ z. B. Kantenrelation eines Graphen: Adjazenzmatrix

Das sollten Sie üben:

- ▶ zu gegebenem Graphen die Adjazenzmatrix hinschreiben
- ▶ zu gegebener Adjazenzmatrix den Graphen hinmalen
- ▶ z. B. für irgendwelche „speziellen“ Graphen und Matrizen

Repräsentation von Graphen im Rechner

Berechnung der 2-Erreichbarkeitsrelation und Rechnen mit Matrizen

- 2-Erreichbarkeit an einem Beispiel

- Matrixmultiplikation

- Matrixaddition

Einfache Berechnung der Erreichbarkeitsrelation

- Potenzen der Adjazenzmatrix

- Erste Möglichkeit für die Berechnung der Wegematrix

- Zählen durchzuführender arithmetischer Operationen

- Weitere Möglichkeiten für die Berechnung der Wegematrix

Algorithmus von Warshall

Repräsentation von Graphen im Rechner

Berechnung der 2-Erreichbarkeitsrelation und Rechnen mit Matrizen

2-Erreichbarkeit an einem Beispiel

Matrixmultiplikation

Matrixaddition

Einfache Berechnung der Erreichbarkeitsrelation

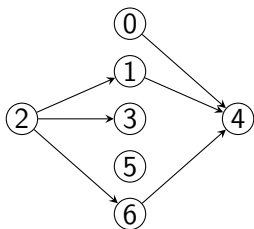
Potenzen der Adjazenzmatrix

Erste Möglichkeit für die Berechnung der Wegematrix

Zählen durchzuführender arithmetischer Operationen

Weitere Möglichkeiten für die Berechnung der Wegematrix

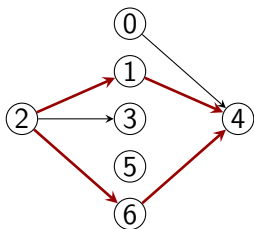
Algorithmus von Warshall



$$\begin{array}{c} \begin{matrix} & 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

- von Interesse: Pfade der Länge 2 von Knoten 2 zu Knoten 4

Ein Beispielgraph



$$\begin{array}{c} \begin{matrix} & 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

- ▶ von Interesse: Pfade der Länge 2 von Knoten 2 zu Knoten 4
- ▶ hinsehen: $(2, 1, 4)$ und $(2, 1, 6)$

- ▶ Wie findet man „systematisch“ alle interessierenden Pfade?
- ▶ prüfe *alle* Knoten $k \in V$:
 - ▶ Ist $(2, k, 4)$ ein Pfad?
 - ▶ Ist $(2, k) \in E$ und $(k, 4) \in E$?
 - ▶ Ist $A_{2k} = 1$ und $A_{k4} = 1$?
 - ▶ Ist $A_{2k} \cdot A_{k4} = 1$?
- ▶ durchlaufe nacheinander parallel
 - ▶ alle A_{2k} und alle A_{k4} , d. h.
 - ▶ *Zeile* für Knoten 2 und *Spalte* für Knoten 4

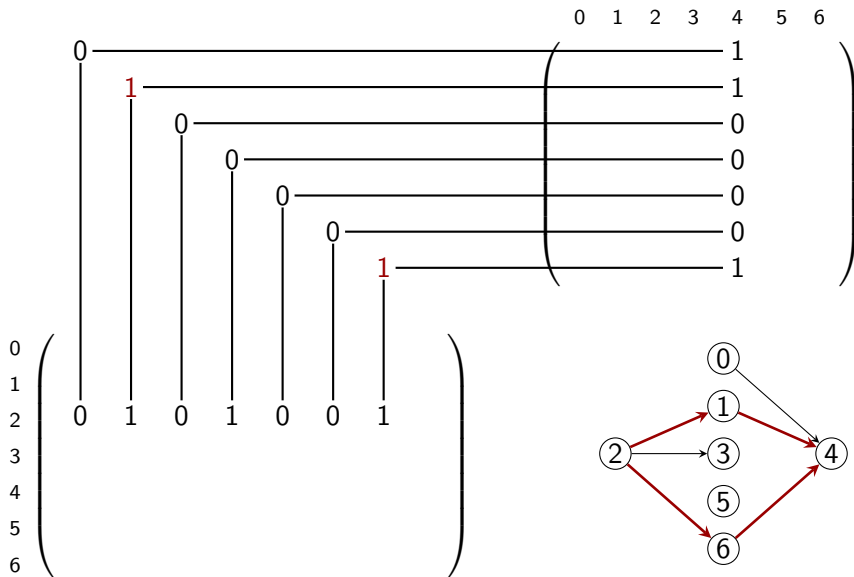
- ▶ Wie findet man „systematisch“ alle interessierenden Pfade?
- ▶ prüfe *alle* Knoten $k \in V$:
 - ▶ Ist $(2, k, 4)$ ein Pfad?
 - ▶ Ist $(2, k) \in E$ und $(k, 4) \in E$?
 - ▶ Ist $A_{2k} = 1$ und $A_{k4} = 1$?
 - ▶ Ist $A_{2k} \cdot A_{k4} = 1$?
- ▶ durchlaufe nacheinander parallel
 - ▶ alle A_{2k} und alle A_{k4} , d. h.
 - ▶ *Zeile* für Knoten 2 und *Spalte* für Knoten 4

- ▶ Wie findet man „systematisch“ alle interessierenden Pfade?
- ▶ prüfe *alle* Knoten $k \in V$:
 - ▶ Ist $(2, k, 4)$ ein Pfad?
 - ▶ Ist $(2, k) \in E$ und $(k, 4) \in E$?
 - ▶ Ist $A_{2k} = 1$ und $A_{k4} = 1$?
 - ▶ Ist $A_{2k} \cdot A_{k4} = 1$?
- ▶ durchlaufe nacheinander parallel
 - ▶ alle A_{2k} und alle A_{k4} , d. h.
 - ▶ *Zeile* für Knoten 2 und *Spalte* für Knoten 4

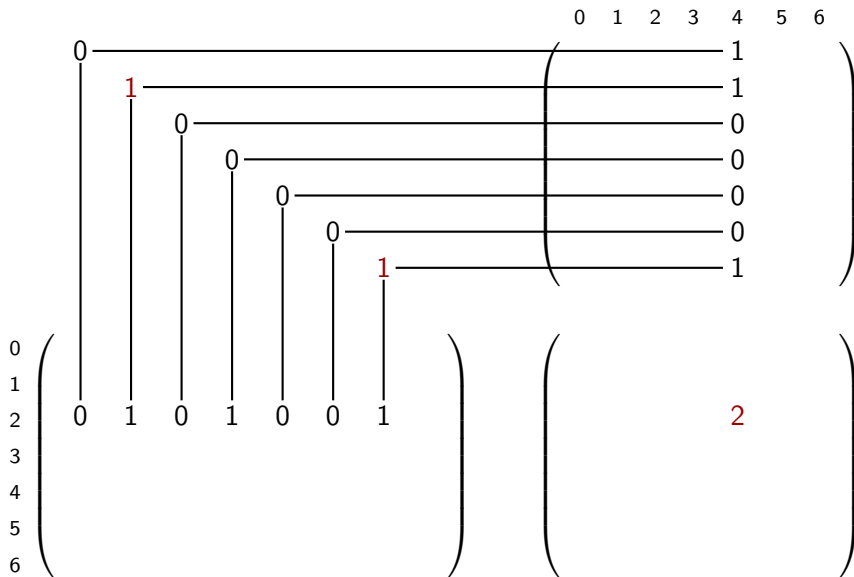
- ▶ Wie findet man „systematisch“ alle interessierenden Pfade?
- ▶ prüfe *alle* Knoten $k \in V$:
 - ▶ Ist $(2, k, 4)$ ein Pfad?
 - ▶ Ist $(2, k) \in E$ und $(k, 4) \in E$?
 - ▶ Ist $A_{2k} = 1$ und $A_{k4} = 1$?
 - ▶ Ist $A_{2k} \cdot A_{k4} = 1$?
- ▶ durchlaufe nacheinander parallel
 - ▶ alle A_{2k} und alle A_{k4} , d. h.
 - ▶ *Zeile* für Knoten 2 und *Spalte* für Knoten 4

- ▶ Wie findet man „systematisch“ alle interessierenden Pfade?
- ▶ prüfe *alle* Knoten $k \in V$:
 - ▶ Ist $(2, k, 4)$ ein Pfad?
 - ▶ Ist $(2, k) \in E$ und $(k, 4) \in E$?
 - ▶ Ist $A_{2k} = 1$ und $A_{k4} = 1$?
 - ▶ Ist $A_{2k} \cdot A_{k4} = 1$?
- ▶ durchlaufe nacheinander parallel
 - ▶ alle A_{2k} und alle A_{k4} , d. h.
 - ▶ *Zeile* für Knoten 2 und *Spalte* für Knoten 4

Beispielgraph: systematische Suche nach Pfaden



Beispielgraph: Zählen der Pfade



Beispielgraph: Zählen der Pfade (2)

$$P_{24} = \sum_{k=0}^6 A_{2k} \cdot A_{k4}$$

$$\begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} \begin{pmatrix} & & & & & & \\ & & & & & & \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \end{pmatrix}$$

$$\begin{matrix} & 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \begin{pmatrix} & & & & 1 & & \\ & & & & 1 & & \\ & & & & 0 & & \\ & & & & 0 & & \\ & & & & 0 & & \\ & & & & 0 & & \\ & & & & 0 & & \\ & & & & 1 & & \end{pmatrix}$$

$$\begin{pmatrix} & & & & & & \\ & & & & & & \\ & & & & 2 & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \end{pmatrix}$$

Repräsentation von Graphen im Rechner

Berechnung der 2-Erreichbarkeitsrelation und Rechnen mit Matrizen

2-Erreichbarkeit an einem Beispiel

Matrixmultiplikation

Matrixaddition

Einfache Berechnung der Erreichbarkeitsrelation

Potenzen der Adjazenzmatrix

Erste Möglichkeit für die Berechnung der Wegematrix

Zählen durchzuführender arithmetischer Operationen

Weitere Möglichkeiten für die Berechnung der Wegematrix

Algorithmus von Warshall

- ▶ es sei
 - ▶ A eine $\ell \times n$ -Matrix
 - ▶ B eine $n \times m$ -Matrix
- ▶ die $\ell \times m$ -Matrix C mit

$$C_{ij} = \sum_{k=0}^{n-1} A_{ik} \cdot B_{kj}$$

- ▶ heißt das *Produkt* von A und B
- ▶ geschrieben $C = A \cdot B$
- ▶ **Achtung:** im Allgemeinen $A \cdot B \neq B \cdot A$!



$$C_{ij} = \sum_{k=0}^{n-1} A_{ik} \cdot B_{kj}$$

- ▶ erst mal nur die naheliegende Möglichkeit
- ▶ es geht auch anders!

```
for  $i \leftarrow 0$  to  $\ell - 1$  do  
  for  $j \leftarrow 0$  to  $m - 1$  do  
     $C_{ij} \leftarrow 0$   
    for  $k \leftarrow 0$  to  $n - 1$  do  
       $C_{ij} \leftarrow C_{ij} + A_{ik} \cdot B_{kj}$   
    od  
  od  
od
```

- ▶ *Einheitsmatrix*: $n \times n$ -Matrix I , bei der für alle i und j gilt:

$$I_{ij} = \begin{cases} 1 & \text{falls } i = j \\ 0 & \text{falls } i \neq j \end{cases}$$

- ▶ für jede $m \times n$ -Matrix A gilt:

$$I \cdot A = A = A \cdot I$$

- ▶ Beachte: verschiedene Größen der Einheitsmatrizen links und rechts

$$A^0 = I$$

$$\forall n \in \mathbb{N}_0 : A^{n+1} = A^n \cdot A$$

- ▶ Quadrat der Adjazenzmatrix A enthält nach Definition der Matrixmultiplikation als Eintrag in Zeile i und Spalte j

$$(A^2)_{ij} = \sum_{k=0}^{n-1} A_{ik} A_{kj} .$$

- ▶ Jeder Summand $A_{ik} A_{kj}$ ist 1 gdw.
 - ▶ $A_{ik} = A_{kj} = 1$ ist, also gdw.
 - ▶ Kanten von i nach k und von k nach j existieren, also gdw.
 - ▶ (i, k, j) ein Pfad der Länge 2 von i nach j ist.
- und 0 sonst.
- ▶ Für $k_1 \neq k_2$ sind (i, k_1, j) und (i, k_2, j) verschiedene Pfade.
- ▶ Also ist

$$(A^2)_{ij} = \sum_{k=0}^{n-1} A_{ik} A_{kj}$$

gleich der Anzahl der Pfade der Länge 2 von i nach j .

Repräsentation von Graphen im Rechner

Berechnung der 2-Erreichbarkeitsrelation und Rechnen mit Matrizen

2-Erreichbarkeit an einem Beispiel

Matrixmultiplikation

Matrixaddition

Einfache Berechnung der Erreichbarkeitsrelation

Potenzen der Adjazenzmatrix

Erste Möglichkeit für die Berechnung der Wegematrix

Zählen durchzuführender arithmetischer Operationen

Weitere Möglichkeiten für die Berechnung der Wegematrix

Algorithmus von Warshall

- ▶ es seien A und B zwei $m \times n$ -Matrizen
- ▶ die $m \times n$ -Matrix C mit

$$C_{ij} = A_{ij} + B_{ij}$$

- ▶ heißt die *Summe* von A und B
- ▶ geschrieben $C = A + B$
- ▶ stets $A + B = B + A$
- ▶ neutrales Element: die *Nullmatrix*, die überall Nullen enthält
- ▶ geschrieben 0
- ▶ algorithmisch:
 for $i \leftarrow 0$ **to** $m - 1$ **do**
 for $j \leftarrow 0$ **to** $n - 1$ **do**
 $C_{ij} \leftarrow A_{ij} + B_{ij}$
 od
 od

Repräsentation von Graphen im Rechner

Berechnung der 2-Erreichbarkeitsrelation und Rechnen mit Matrizen

2-Erreichbarkeit an einem Beispiel

Matrixmultiplikation

Matrixaddition

Einfache Berechnung der Erreichbarkeitsrelation

Potenzen der Adjazenzmatrix

Erste Möglichkeit für die Berechnung der Wegematrix

Zählen durchzuführender arithmetischer Operationen

Weitere Möglichkeiten für die Berechnung der Wegematrix

Algorithmus von Warshall

- ▶ Benutze

$$E^* = \bigcup_{i=0}^{\infty} E^i$$

um die Wegematrix zu berechnen.

- ▶ Probleme:

- ▶ Was kann man gegen das unendlich tun?
- ▶ Woher kommen die Matrizen für die Relationen E^i ?
- ▶ Welcher Matrizen-Operation entspricht die Vereinigung?

- ▶ Benutze

$$E^* = \bigcup_{i=0}^{\infty} E^i$$

um die Wegematrix zu berechnen.

- ▶ Probleme:

- ▶ Was kann man gegen das **unendlich** tun?
- ▶ Woher kommen die Matrizen für die Relationen E^i ?
- ▶ Welcher Matrizen-Operation entspricht die Vereinigung?

- ▶ Benutze

$$E^* = \bigcup_{i=0}^{\infty} E^i$$

um die Wegematrix zu berechnen.

- ▶ Probleme:
 - ▶ Was kann man gegen das unendlich tun?
 - ▶ Woher kommen die Matrizen für die Relationen E^i ?
 - ▶ Welcher Matrizen-Operation entspricht die Vereinigung?

- ▶ Benutze

$$E^* = \bigcup_{i=0}^{\infty} E^i$$

um die Wegematrix zu berechnen.

- ▶ Probleme:
 - ▶ Was kann man gegen das unendlich tun?
 - ▶ Woher kommen die Matrizen für die Relationen E^i ?
 - ▶ Welcher Matrizen-Operation entspricht die **Vereinigung**?

- ▶ bei Graphen spezieller: nur *endlich* viele Knoten
- ▶ Frage: Existiert ein Pfad in G von Knoten i nach Knoten j ?
- ▶ Sei
 - ▶ $G = (V, E)$ mit $|V| = n$ Knoten.
 - ▶ $p = (i_0, i_1, \dots, i_k)$ ein Pfad mit $i_0 = i$ und $i_k = j$.
- ▶ Wenn $k \geq n$, dann
 - ▶ kommen in der Liste p also $k + 1 \geq n + 1$ „Knotennamen“ vor.
 - ▶ aber G hat nur n verschiedene Knoten.
 - ▶ Also muss ein Knoten x doppelt in der Liste p vorkommen.
 - ▶ p enthält Zyklus von x nach x
- ▶ Weglassen des Zyklus
 - ▶ ergibt kürzeren Pfad,
 - ▶ der immer noch von i nach j führt.
- ▶ wiederhole, solange Pfad mindestens $n + 1$ Knoten enthält
- ▶ Ergebnis: Pfad, in dem höchstens noch n Knoten, also höchstens $n - 1$ Kanten, vorkommen, und der auch immer noch von i nach j führt.

- ▶ bei Graphen spezieller: nur *endlich* viele Knoten
- ▶ Frage: Existiert ein Pfad in G von Knoten i nach Knoten j ?
- ▶ Sei
 - ▶ $G = (V, E)$ mit $|V| = n$ Knoten.
 - ▶ $p = (i_0, i_1, \dots, i_k)$ ein Pfad mit $i_0 = i$ und $i_k = j$.
- ▶ Wenn $k \geq n$, dann
 - ▶ kommen in der Liste p also $k + 1 \geq n + 1$ „Knotennamen“ vor.
 - ▶ aber G hat nur n verschiedene Knoten.
 - ▶ Also muss ein Knoten x doppelt in der Liste p vorkommen.
 - ▶ p enthält Zyklus von x nach x
- ▶ Weglassen des Zyklus
 - ▶ ergibt kürzeren Pfad,
 - ▶ der immer noch von i nach j führt.
- ▶ wiederhole, solange Pfad mindestens $n + 1$ Knoten enthält
- ▶ Ergebnis: Pfad, in dem höchstens noch n Knoten, also höchstens $n - 1$ Kanten, vorkommen, und der auch immer noch von i nach j führt.

- ▶ bei Graphen spezieller: nur *endlich* viele Knoten
- ▶ Frage: Existiert ein Pfad in G von Knoten i nach Knoten j ?
- ▶ Sei
 - ▶ $G = (V, E)$ mit $|V| = n$ Knoten.
 - ▶ $p = (i_0, i_1, \dots, i_k)$ ein Pfad mit $i_0 = i$ und $i_k = j$.
- ▶ Wenn $k \geq n$, dann
 - ▶ kommen in der Liste p also $k + 1 \geq n + 1$ „Knotennamen“ vor.
 - ▶ aber G hat nur n verschiedene Knoten.
 - ▶ Also muss ein Knoten x doppelt in der Liste p vorkommen.
 - ▶ p enthält Zyklus von x nach x
- ▶ Weglassen des Zyklus
 - ▶ ergibt kürzeren Pfad,
 - ▶ der immer noch von i nach j führt.
- ▶ wiederhole, solange Pfad mindestens $n + 1$ Knoten enthält
- ▶ Ergebnis: Pfad, in dem höchstens noch n Knoten, also höchstens $n - 1$ Kanten, vorkommen, und der auch immer noch von i nach j führt.

- ▶ bei Graphen spezieller: nur *endlich* viele Knoten
- ▶ Frage: Existiert ein Pfad in G von Knoten i nach Knoten j ?
- ▶ Sei
 - ▶ $G = (V, E)$ mit $|V| = n$ Knoten.
 - ▶ $p = (i_0, i_1, \dots, i_k)$ ein Pfad mit $i_0 = i$ und $i_k = j$.
- ▶ Wenn $k \geq n$, dann
 - ▶ kommen in der Liste p also $k + 1 \geq n + 1$ „Knotennamen“ vor.
 - ▶ aber G hat nur n verschiedene Knoten.
 - ▶ Also muss ein Knoten x doppelt in der Liste p vorkommen.
 - ▶ p enthält Zyklus von x nach x
- ▶ Weglassen des Zyklus
 - ▶ ergibt kürzeren Pfad,
 - ▶ der immer noch von i nach j führt.
- ▶ wiederhole, solange Pfad mindestens $n + 1$ Knoten enthält
- ▶ Ergebnis: Pfad, in dem höchstens noch n Knoten, also höchstens $n - 1$ Kanten, vorkommen, und der auch immer noch von i nach j führt.

- ▶ bei Graphen spezieller: nur *endlich* viele Knoten
- ▶ Frage: Existiert ein Pfad in G von Knoten i nach Knoten j ?
- ▶ Sei
 - ▶ $G = (V, E)$ mit $|V| = n$ Knoten.
 - ▶ $p = (i_0, i_1, \dots, i_k)$ ein Pfad mit $i_0 = i$ und $i_k = j$.
- ▶ Wenn $k \geq n$, dann
 - ▶ kommen in der Liste p also $k + 1 \geq n + 1$ „Knotennamen“ vor.
 - ▶ aber G hat nur n verschiedene Knoten.
 - ▶ Also muss ein Knoten x doppelt in der Liste p vorkommen.
 - ▶ p enthält Zyklus von x nach x
- ▶ Weglassen des Zyklus
 - ▶ ergibt kürzeren Pfad,
 - ▶ der immer noch von i nach j führt.
- ▶ wiederhole, solange Pfad mindestens $n + 1$ Knoten enthält
- ▶ Ergebnis: Pfad, in dem höchstens noch n Knoten, also höchstens $n - 1$ Kanten, vorkommen, und der auch immer noch von i nach j führt.

Beseitigung der *unendlichen* Vereinigung (2)

Für Erreichbarkeit in einem endlichen Graphen mit n Knoten angeht, gilt:

$$E^* = \bigcup_{i=0}^{n-1} E^i$$

Betrachtung höherer Potenzen (längerer Pfade) schadet nicht:

Lemma

Für jeden gerichteten Graphen $G = (V, E)$ mit n Knoten gilt:

$$\forall k \geq n - 1 : E^* = \bigcup_{i=0}^k E^i$$

Für Erreichbarkeit in einem endlichen Graphen mit n Knoten angeht, gilt:

$$E^* = \bigcup_{i=0}^{n-1} E^i$$

Betrachtung höherer Potenzen (längerer Pfade) schadet nicht:

Lemma

Für jeden gerichteten Graphen $G = (V, E)$ mit n Knoten gilt:

$$\forall k \geq n - 1 : E^* = \bigcup_{i=0}^k E^i$$

Repräsentation von Graphen im Rechner

Berechnung der 2-Erreichbarkeitsrelation und Rechnen mit Matrizen

2-Erreichbarkeit an einem Beispiel

Matrixmultiplikation

Matrixaddition

Einfache Berechnung der Erreichbarkeitsrelation

Potenzen der Adjazenzmatrix

Erste Möglichkeit für die Berechnung der Wegematrix

Zählen durchzuführender arithmetischer Operationen

Weitere Möglichkeiten für die Berechnung der Wegematrix

Algorithmus von Warshall

Lemma

Es sei G ein gerichteter Graph mit Adjazenzmatrix A .

Für alle $k \in \mathbb{N}_0$ gilt:

$(A^k)_{ij}$ ist die Anzahl der Pfade der Länge k in G von i nach j .

- ▶ Beweis durch vollständige Induktion.
- ▶ Induktionsschritt fast wie im Fall $k = 2$.

- ▶ *Signum-Funktion*

$$\operatorname{sgn} : \mathbb{R} \rightarrow \mathbb{R} : \operatorname{sgn}(x) = \begin{cases} 1 & \text{falls } x > 0 \\ 0 & \text{falls } x = 0 \\ -1 & \text{falls } x < 0 \end{cases}$$

- ▶ Erweiterung auf Matrizen durch komponentenweise Anwendung

$$\operatorname{sgn} : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n} : (\operatorname{sgn}(M))_{ij} = \operatorname{sgn}(M_{ij})$$

Korollar

Es sei G ein gerichteter Graph mit Adjazenzmatrix A .

Für alle $k \in \mathbb{N}_0$ gilt:

1.

$$\text{sgn}((A^k)_{ij}) = \begin{cases} 1 & \text{falls in } G \text{ ein Pfad der Länge } k \\ & \text{von } i \text{ nach } j \text{ existiert} \\ 0 & \text{falls in } G \text{ kein Pfad der Länge } k \\ & \text{von } i \text{ nach } j \text{ existiert} \end{cases}$$

2. Matrix $\text{sgn}(A^k)$ repräsentiert die Relation E^k .

Repräsentation von Graphen im Rechner

Berechnung der 2-Erreichbarkeitsrelation und Rechnen mit Matrizen

2-Erreichbarkeit an einem Beispiel

Matrixmultiplikation

Matrixaddition

Einfache Berechnung der Erreichbarkeitsrelation

Potenzen der Adjazenzmatrix

Erste Möglichkeit für die Berechnung der Wegematrix

Zählen durchzuführender arithmetischer Operationen

Weitere Möglichkeiten für die Berechnung der Wegematrix

Algorithmus von Warshall

- ▶ Seien Relationen $R \subseteq M \times M$ und $R' \subseteq M \times M$ repräsentiert durch Matrizen A und A' .
- ▶ dann:

$$\begin{aligned}(i, j) \in R \cup R' &\iff (i, j) \in R \vee (i, j) \in R' \\ &\iff A_{ij} = 1 \vee A'_{ij} = 1 \\ &\iff A_{ij} + A'_{ij} \geq 1 \\ &\iff (A + A')_{ij} \geq 1 \\ &\iff \text{sgn}(A + A')_{ij} = 1\end{aligned}$$

- ▶ also: $R \cup R'$ wird durch $\text{sgn}(A + A')$ repräsentiert.

Lemma

Es sei G ein gerichteter Graph mit Adjazenzmatrix A .

Dann gilt für alle $k \geq n - 1$:

- ▶ *Die Matrix $\text{sgn}(\sum_{i=0}^k A^i)$ repräsentiert die Relation E^* .*
- ▶ *Mit anderen Worten:*

$$W = \text{sgn} \left(\sum_{i=0}^k A^i \right)$$

ist die Wegematrix des Graphen G .

Man muss sich noch überlegen:

- ▶ $\bigcup_{i=0}^{n-1} E^i$ wird durch Matrix $\text{sgn}(\sum_{i=0}^k \text{sgn}(A^i))$ repräsentiert.
 - ▶ leichte Verallgemeinerung des Falles $R \cup R'$
- ▶ In dieser Formel darf man die „inneren“ Anwendungen von sgn weglassen.
 - ▶ Wenn alle Matrixeinträge ≥ 0 sind, gilt:

$$\text{sgn}(\text{sgn}(M) + \text{sgn}(M'))_{ij} = \text{sgn}(M + M')_{ij}$$

Einfachster Algorithmus für die Wegematrix

```
// Matrix  $A$  sei die Adjazenzmatrix
// Matrix  $W$  wird am Ende die Wegematrix enthalten
// Matrix  $M$  wird benutzt um  $A^i$  zu berechnen
 $W \leftarrow 0$  // Nullmatrix
for  $i \leftarrow 0$  to  $n - 1$  do
     $M \leftarrow I$  // Einheitsmatrix
    for  $j \leftarrow 1$  to  $i$  do
         $M \leftarrow M \cdot A$  // Matrixmultiplikation
    od
     $W \leftarrow W + M$  // Matrixaddition
od
 $W \leftarrow \text{sgn}(W)$ 
```

Repräsentation von Graphen im Rechner

Berechnung der 2-Erreichbarkeitsrelation und Rechnen mit Matrizen

2-Erreichbarkeit an einem Beispiel

Matrixmultiplikation

Matrixaddition

Einfache Berechnung der Erreichbarkeitsrelation

Potenzen der Adjazenzmatrix

Erste Möglichkeit für die Berechnung der Wegematrix

Zählen durchzuführender arithmetischer Operationen

Weitere Möglichkeiten für die Berechnung der Wegematrix

Algorithmus von Warshall

Was ist der „Aufwand“ eines Algorithmus?

- ▶ Anzahl Codezeilen?
- ▶ Entwicklungszeit?
- ▶ Anzahl Schritte?
 - ▶ nicht immer gleich
- ▶ benötigter Speicherplatz?
 - ▶ nicht immer gleich
- ▶ vorläufig(!): Anzahl arithmetischer Operationen

Wieviele elementare Operationen für Matrixaddition?

- ▶ Matrixaddition:

```
for  $i \leftarrow 0$  to  $m - 1$  do
  for  $j \leftarrow 0$  to  $n - 1$  do
     $C_{ij} \leftarrow A_{ij} + B_{ij}$ 
  od
od
```

- ▶ $m \cdot n$ Additionen
- ▶ für $n \times n$ -Matrizen: n^2

Wieviele elementare Operationen für Matrixmultiplikation?

- ▶ Matrixmultiplikation

```
for  $i \leftarrow 0$  to  $\ell - 1$  do
  for  $j \leftarrow 0$  to  $m - 1$  do
     $C_{ij} \leftarrow 0$ 
    for  $k \leftarrow 0$  to  $n - 1$  do
       $C_{ij} \leftarrow C_{ij} + A_{ik} \cdot B_{kj}$ 
    od
  od
od
```

- ▶ $\ell \cdot m \cdot n$ Additionen und $\ell \cdot m \cdot n$ Multiplikationen
- ▶ kleine Variante: $\ell \cdot m \cdot (n - 1)$ Additionen
- ▶ insgesamt für $n \times n$ -Matrizen: $2n^3$ bzw. $2n^3 - n^2$
- ▶ **Achtung:** Niemand sagt, dass das die einzige oder gar beste Methode ist. Sie ist es nicht!

Wieviele elementare Operationen für Wegematrix?

► Algorithmus:

```
W ← 0
for i ← 0 to n - 1 do
  M ← I
  for j ← 1 to i do
    M ← M · A
  od
  W ← W + M
od
W ← sgn(W)
```

► Aufwand:

$$\left(\sum_{i=0}^{n-1} i \right) \cdot (2n^3 - n^2) + n \cdot n^2 + n^2 = n^5 - \frac{3}{2}n^4 + \frac{3}{2}n^3 + n^2$$

Wieviele elementare Operationen für Wegematrix?

► Algorithmus:

```
W ← 0
for i ← 0 to n - 1 do
  M ← I
  for j ← 1 to i do
    M ← M · A
  od
  W ← W + M
od
W ← sgn(W)
```

► Aufwand:

$$\left(\sum_{i=0}^{n-1} i \right) \cdot (2n^3 - n^2) + n \cdot n^2 + n^2 = n^5 - \frac{3}{2}n^4 + \frac{3}{2}n^3 + n^2$$

Wieviele elementare Operationen für Wegematrix?

► Algorithmus:

```
W ← 0
for i ← 0 to n - 1 do
  M ← I
  for j ← 1 to i do
    M ← M · A
  od
  W ← W + M
od
W ← sgn(W)
```

► Aufwand:

$$\left(\sum_{i=0}^{n-1} i \right) \cdot (2n^3 - n^2) + n \cdot n^2 + n^2 = n^5 - \frac{3}{2}n^4 + \frac{3}{2}n^3 + n^2$$

Wieviele elementare Operationen für Wegematrix?

► Algorithmus:

```
W ← 0
for i ← 0 to n - 1 do
  M ← I
  for j ← 1 to i do
    M ← M · A
  od
  W ← W + M
od
W ← sgn(W)
```

► Aufwand:

$$\left(\sum_{i=0}^{n-1} i \right) \cdot (2n^3 - n^2) + n \cdot n^2 + n^2 = n^5 - \frac{3}{2}n^4 + \frac{3}{2}n^3 + n^2$$

Wieviele elementare Operationen für Wegematrix?

► Algorithmus:

```
W ← 0
for i ← 0 to n - 1 do
  M ← I
  for j ← 1 to i do
    M ← M · A
  od
  W ← W + M
od
W ← sgn(W)
```

► Aufwand:

$$\left(\sum_{i=0}^{n-1} i \right) \cdot (2n^3 - n^2) + n \cdot n^2 + n^2 = n^5 - \frac{3}{2}n^4 + \frac{3}{2}n^3 + n^2$$

Wieviele elementare Operationen für Wegematrix?

► Algorithmus:

```
W ← 0
for i ← 0 to n - 1 do
  M ← I
  for j ← 1 to i do
    M ← M · A
  od
  W ← W + M
od
W ← sgn(W)
```

► Aufwand:

$$\left(\sum_{i=0}^{n-1} i \right) \cdot (2n^3 - n^2) + n \cdot n^2 + n^2 = n^5 - \frac{3}{2}n^4 + \frac{3}{2}n^3 + n^2$$

Repräsentation von Graphen im Rechner

Berechnung der 2-Erreichbarkeitsrelation und Rechnen mit Matrizen

2-Erreichbarkeit an einem Beispiel

Matrixmultiplikation

Matrixaddition

Einfache Berechnung der Erreichbarkeitsrelation

Potenzen der Adjazenzmatrix

Erste Möglichkeit für die Berechnung der Wegematrix

Zählen durchzuführender arithmetischer Operationen

Weitere Möglichkeiten für die Berechnung der Wegematrix

Algorithmus von Warshall

Da kann man etwas besser machen!

- ▶ haben so getan, als wären für A^i immer $i - 1$ Matrixmultiplikationen nötig
- ▶ es werden aber ohnehin *alle* Potenzen A^i benötigt
- ▶ also besser immer das alte A^{i-1} merken und wiederverwenden
- ▶ Algorithmus:

$W \leftarrow 0$

$M \leftarrow I$

for $i \leftarrow 0$ **to** $n - 1$ **do**

$W \leftarrow W + M$

$M \leftarrow M \cdot A$

od

$W \leftarrow \text{sgn}(W)$

- ▶ Aufwand:

$$n \cdot (n^2 + (2n^3 - n^2)) + n^2 = 2n^4 + n^2$$

- ▶ Schon vergessen?

$$\forall k \geq n - 1 : E^* = \bigcup_{i=0}^k E^i$$

Alle $k \geq n - 1$ sind in Ordnung.

- ▶ Aber warum kann das helfen?
 - ▶ wählen statt $n - 1$ kleinste Zweierpotenz $k = 2^m \geq n$, also $m = \lceil \log_2 n \rceil$
 - ▶ finden eine Matrix F mit $W = F^{2^m} = (\dots ((F^2)^2) \dots)^2$
 - ▶ Das sind nur noch $m = \lceil \log_2 n \rceil$ Matrixmultiplikationen!
- ▶ Preisfrage: Wie sieht F aus?
- ▶ Antwort: Wähle $F = E^0 \cup E^1 = I_V \cup E$.

- ▶ Schon vergessen?

$$\forall k \geq n - 1 : E^* = \bigcup_{i=0}^k E^i$$

Alle $k \geq n - 1$ sind in Ordnung.

- ▶ Aber warum kann das helfen?
 - ▶ wählen statt $n - 1$ kleinste Zweierpotenz $k = 2^m \geq n$, also $m = \lceil \log_2 n \rceil$
 - ▶ finden eine Matrix F mit $W = F^{2^m} = (\dots ((F^2)^2) \dots)^2$
 - ▶ Das sind nur noch $m = \lceil \log_2 n \rceil$ Matrixmultiplikationen!
- ▶ Preisfrage: Wie sieht F aus?
- ▶ Antwort: Wähle $F = E^0 \cup E^1 = I_V \cup E$.

- ▶ Schon vergessen?

$$\forall k \geq n - 1 : E^* = \bigcup_{i=0}^k E^i$$

Alle $k \geq n - 1$ sind in Ordnung.

- ▶ Aber warum kann das helfen?
 - ▶ wählen statt $n - 1$ kleinste Zweierpotenz $k = 2^m \geq n$, also $m = \lceil \log_2 n \rceil$
 - ▶ finden eine Matrix F mit $W = F^{2^m} = (\dots ((F^2)^2) \dots)^2$
 - ▶ Das sind nur noch $m = \lceil \log_2 n \rceil$ Matrixmultiplikationen!
- ▶ Preisfrage: Wie sieht F aus?
- ▶ Antwort: Wähle $F = E^0 \cup E^1 = I_V \cup E$.

- ▶ Schon vergessen?

$$\forall k \geq n - 1 : E^* = \bigcup_{i=0}^k E^i$$

Alle $k \geq n - 1$ sind in Ordnung.

- ▶ Aber warum kann das helfen?
 - ▶ wählen statt $n - 1$ kleinste Zweierpotenz $k = 2^m \geq n$, also $m = \lceil \log_2 n \rceil$
 - ▶ finden eine Matrix F mit $W = F^{2^m} = (\dots ((F^2)^2) \dots)^2$
 - ▶ Das sind nur noch $m = \lceil \log_2 n \rceil$ Matrixmultiplikationen!
- ▶ Preisfrage: Wie sieht F aus?
- ▶ Antwort: Wähle $F = E^0 \cup E^1 = I_V \cup E$.

Es geht noch besser! (2)

► Sei $F = E^0 \cup E^1$

► dann

$$F^2 = (E^0 \cup E^1) \circ (E^0 \cup E^1) = E^0 \cup E^1 \cup E^1 \cup E^2 = E^0 \cup E^1 \cup E^2$$

► und

$$\begin{aligned} F^4 &= (F^2)^2 = (E^0 \cup E^1 \cup E^2) \circ (E^0 \cup E^1 \cup E^2) \\ &= \dots \\ &= E^0 \cup E^1 \cup E^2 \cup E^3 \cup E^4 \end{aligned}$$

► per Induktion: Für alle $m \in \mathbb{N}_0$ gilt:

$$F^{2^m} = \bigcup_{i=0}^{2^m} E^i$$

► Algorithmus:

```
W ← A + I  
m ← ⌈log2 n⌉  
for i ← 1 to m do  
    W ← W · W  
od  
W ← sgn(W)
```

► Aufwand:

$$n^2 + \lceil \log_2 n \rceil + \lceil \log_2 n \rceil \cdot (2n^3 - n^2) + n^2$$

- Beachte: Für die Berechnung des Wertes $\lceil \log_2 n \rceil$ aus n sind höchstens $\lceil \log_2 n \rceil$ Operationen nötig.

Das sollten Sie mitnehmen:

- ▶ Manchmal ist der naheliegende Algorithmus nicht der einzige oder gar der schnellste.
- ▶ Denken/Mathematik/Kreativität/Einfach-mal-drüber-schlafen helfen

Das sollten Sie üben:

- ▶ Aufwandsabschätzungen bei (ineinander geschachtelten) Schleifen
- ▶ auch mal verrückte Ideen ausprobieren

Repräsentation von Graphen im Rechner

Berechnung der 2-Erreichbarkeitsrelation und Rechnen mit Matrizen

- 2-Erreichbarkeit an einem Beispiel

- Matrixmultiplikation

- Matrixaddition

Einfache Berechnung der Erreichbarkeitsrelation

- Potenzen der Adjazenzmatrix

- Erste Möglichkeit für die Berechnung der Wegematrix

- Zählen durchzuführender arithmetischer Operationen

- Weitere Möglichkeiten für die Berechnung der Wegematrix

Algorithmus von Warshall

Der Algorithmus von Warshall

```
for  $i \leftarrow 0$  to  $n - 1$  do  
  for  $j \leftarrow 0$  to  $n - 1$  do  
     $W[i, j] \leftarrow \begin{cases} 1 & \text{falls } i = j \\ A[i, j] & \text{falls } i \neq j \end{cases}$   
  od  
od  
  
for  $k \leftarrow 0$  to  $n - 1$  do  
  for  $i \leftarrow 0$  to  $n - 1$  do  
    for  $j \leftarrow 0$  to  $n - 1$  do  
       $W[i, j] \leftarrow \max( W[i, j], \min( W[i, k], W[k, j] ) )$   
    od  
  od  
od
```


- ▶ algorithmische Idee geht auf eine fundamentale Arbeit von Stephen Kleene zurück
- ▶ Redeweise bei einem Pfad $p = (v_0, v_1, \dots, v_{m-1}, v_m)$ der Länge $m \geq 2$
 - ▶ die Knoten v_1, \dots, v_{m-1} nennen wir **Zwischenknoten** des Pfades.
 - ▶ Pfade der Längen 0 und 1 besitzen keine Zwischenknoten.
- ▶ Invariante für die äußere Schleife

for $k \leftarrow 0$ **to** $n - 1$ **do**

...

od

lautet:

- ▶ Für alle $i, j \in \mathbb{G}_n$: Nach k Durchläufen der äußeren Schleife ist $W[i, j]$ genau dann 1, wenn es einen wiederholungsfreien Pfad von i nach j gibt, bei dem alle Zwischenknoten Nummern in \mathbb{G}_k (also $< k$) haben.

Zum Aufwand des Algorithmus von Warshall

- ▶ drei ineinander geschachtelte Schleifen
- ▶ deren jeweiliger Rumpf n -mal durchlaufen wird
- ▶ „irgendwie ungefähr“ n^3 Operationen

- ▶ Repräsentationen von Graphen im Rechner
- ▶ Berechnung der Wegematrix
 - ▶ mit vielen oder weniger Operationen
 - ▶ Algorithmus Warshall kommt mit weniger Operationen aus als alle unsere vorherigen Versuche