Am Dienstag, den 16. Dezember, ist Eulenfest.

Grundbegriffe der Informatik

Einheit 12: Erste Algorithmen in Graphen

Thomas Worsch

Universität Karlsruhe, Fakultät für Informatik

Wintersemester 2008/2009

Überblick

Java-Arrays und Matrizen

- Eindimensionale Vektoren
- Addition von Vektoren
- Zweidimensionale Matrizen
- Addition von Matrizen
- Multiplikation von Matrizen

Repräsentation von Graphen im Rechner

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

Überblick

Eindimensionales Array

- waagerecht hingemalt:

- senkrecht hingemalt:

- alle Komponenten vom gleichen Typ

Vektoren

- so eine rechteckige Anordnung von Elementen heißt auch *Vektor*, genauer *Zeilenvektor* opder *Spaltenvektor*.
- Die Anzahl der Elemente heißt *Größe* eines Vektors.
- Komponenten oder Einträge des Vektors
  - in Java $V[i]$
  - hier meist auch $V[i]$ (gelegentlich V_i)
- mathematische Notation oft mit großen (runden) Klammern außen herum wie bei

$$\text{Zeilenvektor} \qquad V = \begin{pmatrix} 1 & 2 & 3 & 4 \end{pmatrix}$$

oder

$$\text{Spaltenvektor} \qquad V = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}$$

Addition von Vektoren: mathematisch

- Wenn V_1 und V_2 zwei Vektoren gleicher Größe sind, ist ihre *Summe* definiert
- als der Vektor, bei der für alle Komponentennummern i gilt:

$$(V_1 + V_2)[i] = V_1[i] + V_2[i]$$

- Beispiel:

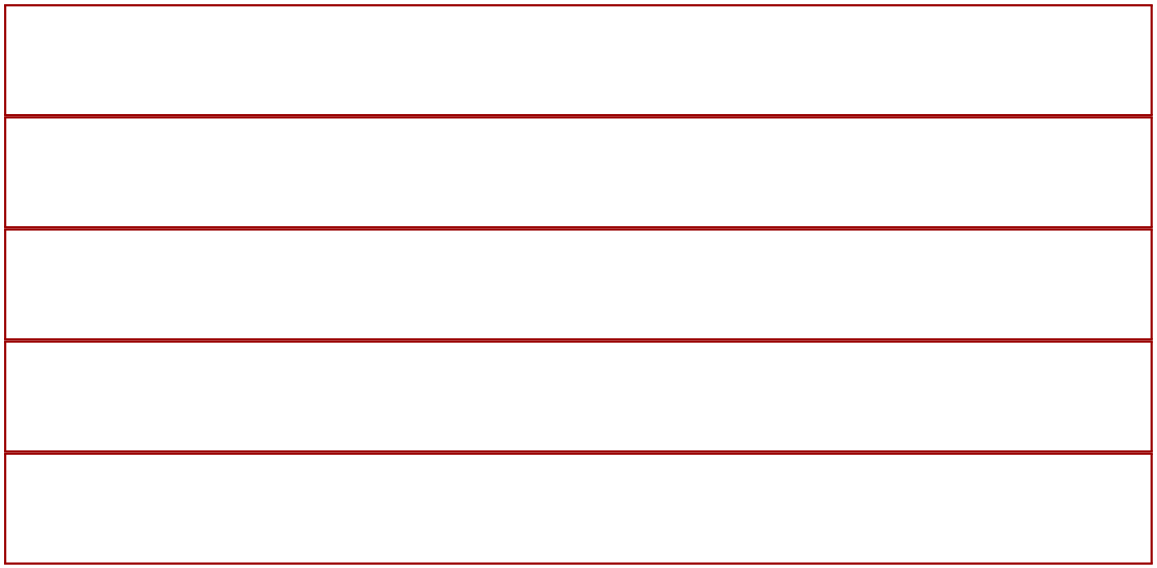
$$\begin{pmatrix}1 & 2 & 3 & 4\end{pmatrix} + \begin{pmatrix}10 & 10 & 10 & 10\end{pmatrix} = \begin{pmatrix}11 & 12 & 13 & 14\end{pmatrix}$$

Addition von Vektoren: algorithmisch

for $i \leftarrow 0$ **to** $n-1$ **do**
  $S[i] \leftarrow V_1[i] + V_2[i]$
od

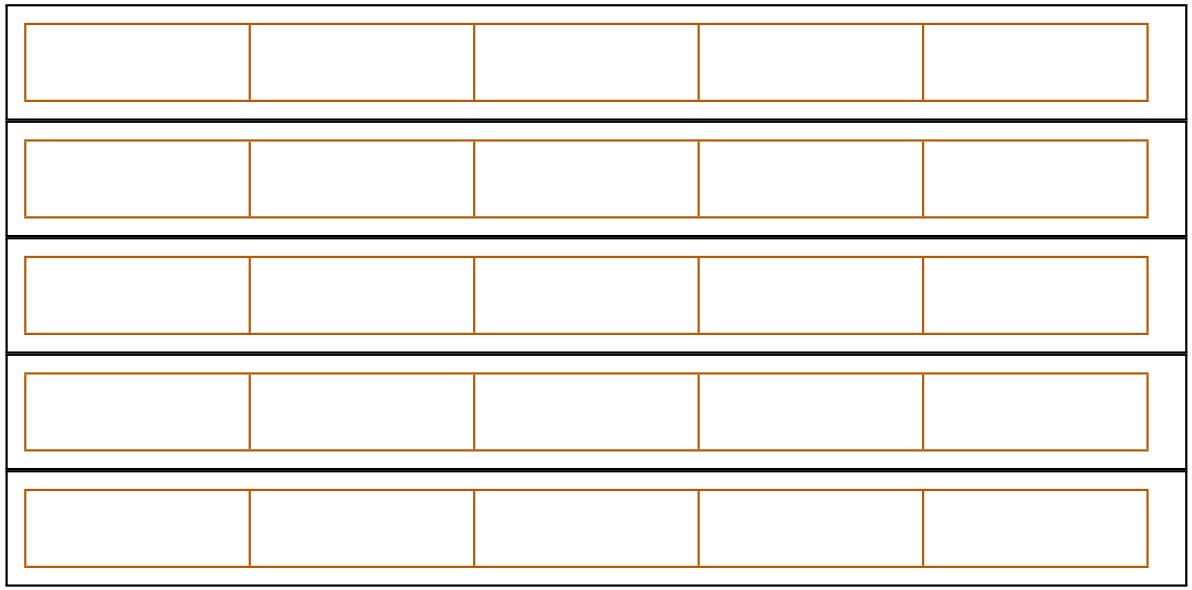
Zweidimensionales Array

eindimensionales Array (senkrecht hingemalt),
dessen Komponenten wieder
eindimensionale Arrays (waagerecht hingemalt) sind



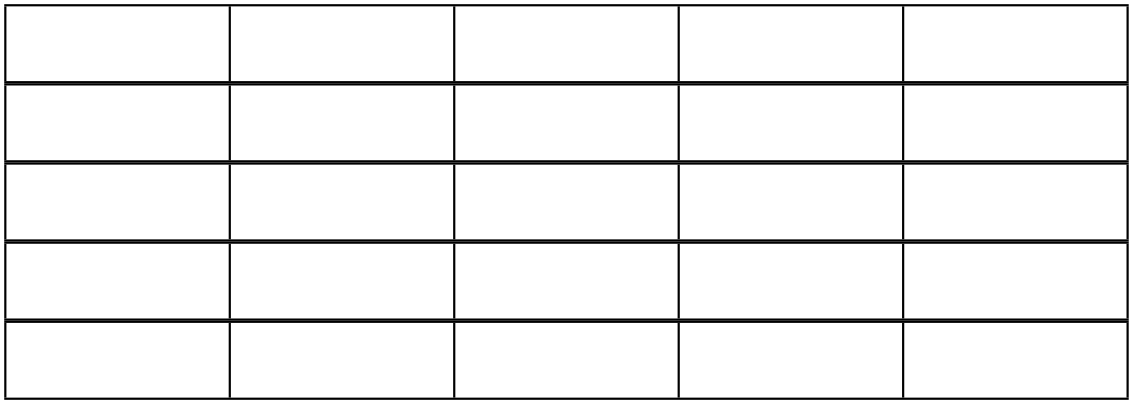
Zweidimensionales Array

eindimensionales Array (senkrecht hingemalt),
dessen Komponenten wieder
eindimensionale Arrays (waagerecht hingemalt) sind



Zweidimensionales Array

eindimensionales Array (senkrecht hingemalt),
dessen Komponenten wieder
eindimensionale Arrays (waagerecht hingemalt) sind



Zweidimensionales Array

eindimensionales Array (senkrecht hingemalt),
dessen Komponenten wieder
eindimensionale Arrays (waagerecht hingemalt) sind

$M[0][0]$	$M[0][1]$	$M[0][2]$	$M[0][3]$	$M[0][4]$
$M[1][0]$	$M[1][1]$	$M[1][2]$	$M[1][3]$	$M[1][4]$
$M[2][0]$	$M[2][1]$	$M[2][2]$	$M[2][3]$	$M[2][4]$
$M[3][0]$	$M[3][1]$	$M[3][2]$	$M[3][3]$	$M[3][4]$
$M[4][0]$	$M[4][1]$	$M[4][2]$	$M[4][3]$	$M[4][4]$

Matrizen

- so eine rechteckige Anordnung von Elementen heißt auch *Matrix*.
- Ihre *Größe* ist das Paar (m, n), bei dem
  - m die Zeilenzahl ist und
  - n die Spaltenzahl
- manchmal auch Schreibweise $m \times n$ für die Größe
- Komponenten oder Einträge der Matrix
  - in Java $M[i][j]$
  - hier lieber $M[i, j]$ oder M_{ij}
- mathematische Notation oft mit großen (runden) Klammern außen herum wie in

$$M = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 11 & 22 & 33 & 44 \\ 111 & 222 & 333 & 444 \end{pmatrix}$$

Addition von Matrizen: mathematisch

- Wenn M_1 und M_2 zwei Matrizen gleicher Größe sind, ist ihre *Summe* definiert
- als die Matrix, bei der für alle Zeilennummern i und alle Spaltennummern j gilt:

$$(M_1 + M_2)[i,j] = M_1[i,j] + M_2[i,j]$$

- Beispiel:

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix} + \begin{pmatrix} 10 & 10 & 10 & 10 \\ 20 & 20 & 20 & 20 \\ 30 & 30 & 30 & 30 \end{pmatrix} = \begin{pmatrix} 11 & 12 & 13 & 14 \\ 21 & 22 & 23 & 24 \\ 31 & 32 & 33 & 34 \end{pmatrix}$$

Addition von Matrizen: algorithmisch

for $i \leftarrow 0$ **to** $m-1$ **do**
  for $j \leftarrow 0$ **to** $n-1$ **do**
    $S[i,j] \leftarrow M_1[i,j] + M_2[i,j]$
  od
od

so viel zur Addition, und nun zur Multiplikation ...

Addition von Matrizen: algorithmisch

for $i \leftarrow 0$ **to** $m-1$ **do**
  for $j \leftarrow 0$ **to** $n-1$ **do**
    $S[i,j] \leftarrow M_1[i,j] + M_2[i,j]$
  od
od

so viel zur Addition, und nun zur Multiplikation ...

Multiplikation von Matrizen: mathematisch

- Wenn M_1 und M_2 zwei Matrizen mit Größen $m_1 \times n_1$ bzw. $m_2 \times n_2$ sind und wenn $n_1 = m_2$ ist, ist ihr *Produkt* definiert
- als die Matrix $M_1 \cdot M_2$ der Größe $m_1 \times n_2$, bei der für alle Zeilennummern i und alle Spaltennummern j gilt:

$$(M_1 \cdot M_2)[i,j] = \sum_{k=0}^{n_1-1} M_1[i,k] \cdot M_2[k,j]$$

- Beispiel:

$$\begin{pmatrix} 3 & 0 & 0 & 1 \\ 0 & -2 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} = M_2$$

$$M_1 = \begin{pmatrix} 0 & 2 & 1 \\ -1 & 1 & 3 \end{pmatrix} \begin{pmatrix} 1 & -3 & 2 & 3 \\ 0 & 1 & 1 & 3 \end{pmatrix} = M_1 \cdot M_2$$

- für iterierte Multiplikation $M \cdots M$ wieder Potenzschreibweise

Multiplikation von Matrizen: mathematisch

- Wenn M_1 und M_2 zwei Matrizen mit Größen $m_1 \times n_1$ bzw. $m_2 \times n_2$ sind und wenn $n_1 = m_2$ ist, ist ihr *Produkt* definiert
- als die Matrix $M_1 \cdot M_2$ der Größe $m_1 \times n_2$, bei der für alle Zeilennummern i und alle Spaltennummern j gilt:

$$(M_1 \cdot M_2)[i,j] = \sum_{k=0}^{n_1-1} M_1[i,k] \cdot M_2[k,j]$$

- Beispiel:

$$\begin{pmatrix} 3 & 0 & 0 & 1 \\ 0 & -2 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} = M_2$$

$$M_1 = \begin{pmatrix} 0 & 2 & 1 \\ -1 & 1 & 3 \end{pmatrix} \begin{pmatrix} 1 & -3 & 2 & 3 \\ 0 & 1 & 1 & 3 \end{pmatrix} = M_1 \cdot M_2$$

- für iterierte Multiplikation $M \cdots M$ wieder Potenzschreibweise

Multiplikation von Matrizen: mathematisch

- Wenn M_1 und M_2 zwei Matrizen mit Größen $m_1 \times n_1$ bzw. $m_2 \times n_2$ sind und wenn $n_1 = m_2$ ist, ist ihr *Produkt* definiert
- als die Matrix $M_1 \cdot M_2$ der Größe $m_1 \times n_2$, bei der für alle Zeilennummern i und alle Spaltennummern j gilt:

$$(M_1 \cdot M_2)[i,j] = \sum_{k=0}^{n_1-1} M_1[i,k] \cdot M_2[k,j]$$

- Beispiel:

$$\begin{pmatrix} 3 & 0 & 0 & 1 \\ 0 & -2 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} = M_2$$

$$M_1 = \begin{pmatrix} 0 & 2 & 1 \\ -1 & 1 & 3 \end{pmatrix} \begin{pmatrix} 1 & -3 & 2 & 3 \\ 0 & 1 & 1 & 3 \end{pmatrix} = M_1 \cdot M_2$$

- für iterierte Multiplikation $M \cdots M$ wieder Potenzschreibweise

Multiplikation von Matrizen: mathematisch

- Wenn M_1 und M_2 zwei Matrizen mit Größen $m_1 \times n_1$ bzw. $m_2 \times n_2$ sind und wenn $n_1 = m_2$ ist, ist ihr *Produkt* definiert
- als die Matrix $M_1 \cdot M_2$ der Größe $m_1 \times n_2$, bei der für alle Zeilennummern i und alle Spaltennummern j gilt:

$$(M_1 \cdot M_2)[i,j] = \sum_{k=0}^{n_1-1} M_1[i,k] \cdot M_2[k,j]$$

- Beispiel:

$$\begin{pmatrix} 3 & 0 & 0 & 1 \\ 0 & -2 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} = M_2$$

$$M_1 = \begin{pmatrix} 0 & 2 & 1 \\ -1 & 1 & 3 \end{pmatrix} \begin{pmatrix} 1 & -3 & 2 & 3 \\ 0 & 1 & 1 & 3 \end{pmatrix} = M_1 \cdot M_2$$

- für iterierte Multiplikation $M \cdots M$ wieder Potenzschreibweise

Multiplikation von Matrizen: mathematisch

- Wenn M_1 und M_2 zwei Matrizen mit Größen $m_1 \times n_1$ bzw. $m_2 \times n_2$ sind und wenn $n_1 = m_2$ ist, ist ihr *Produkt* definiert
- als die Matrix $M_1 \cdot M_2$ der Größe $m_1 \times n_2$, bei der für alle Zeilennummern i und alle Spaltennummern j gilt:

$$(M_1 \cdot M_2)[i,j] = \sum_{k=0}^{n_1-1} M_1[i,k] \cdot M_2[k,j]$$

- Beispiel:

$$\begin{pmatrix} 3 & 0 & 0 & 1 \\ 0 & -2 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} = M_2$$

$$M_1 = \begin{pmatrix} 0 & 2 & 1 \\ -1 & 1 & 3 \end{pmatrix} \begin{pmatrix} 1 & -3 & 2 & 3 \\ 0 & 1 & 1 & 3 \end{pmatrix} = M_1 \cdot M_2$$

- für iterierte Multiplikation $M \cdots M$ wieder Potenzschreibweise

Multiplikation von Matrizen: mathematisch

- Wenn M_1 und M_2 zwei Matrizen mit Größen $m_1 \times n_1$ bzw. $m_2 \times n_2$ sind und wenn $n_1 = m_2$ ist, ist ihr *Produkt* definiert
- als die Matrix $M_1 \cdot M_2$ der Größe $m_1 \times n_2$, bei der für alle Zeilennummern i und alle Spaltennummern j gilt:

$$(M_1 \cdot M_2)[i,j] = \sum_{k=0}^{n_1-1} M_1[i,k] \cdot M_2[k,j]$$

- Beispiel:

$$\begin{pmatrix} 3 & 0 & 0 & 1 \\ 0 & -2 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} = M_2$$

$$M_1 = \begin{pmatrix} 0 & 2 & 1 \\ -1 & 1 & 3 \end{pmatrix} \begin{pmatrix} 1 & -3 & 2 & 3 \\ 0 & 1 & 1 & 3 \end{pmatrix} = M_1 \cdot M_2$$

- für iterierte Multiplikation $M \cdots M$ wieder Potenzschreibweise

Multiplikation mit der Einheitsmatrix

- Einheitsmatrix I der Größe $n \times n$

$$I[i,j] = \begin{cases} 1 & \text{falls } i = j \\ 0 & \text{falls } i \neq j \end{cases}$$

- Beispiel

$$I = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Es ist immer

$$I \cdot M = M = M \cdot I$$

Multiplikation von Matrizen: algorithmisch

- $$P[i,j] = \sum_{k=0}^{n_1-1} M_1[i,k] \cdot M_2[k,j]$$

- ```
for i ← 0 to m1 − 1 do
  for j ← 0 to n2 − 1 do
    P[i, j] ← 0
    for k ← 0 to n1 − 1 do
      P[i, j] ← P[i, j] + M1[i, k] · M2[k, j]
    od
  od
od
```
```

Was ist wichtig

Das sollten Sie mitnehmen:

- zweidimensionale Arrays als „algorithmische“ Objekte
- Matrizen als „mathematische“ Objekte

Das sollten Sie üben:

- Matrizen addieren und multiplizieren

Überblick

Knoten, Kanten und Graphen als Objekte

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

class Edge {
   Vertex start;
   Vertex end;
}

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

Knoten, Kanten und Graphen als Objekte

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

class Edge {
    Vertex start;
    Vertex end;
}

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

Knoten, Kanten und Graphen als Objekte

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

class Edge {
   Vertex start;
   Vertex end;
}

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

Adjazenzlisten

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

class Edge {
   Vertex start;
   Vertex end;
}

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

Inzidenzlisten

```
class Vertex {
    String name;          // oder was auch immer
    Edge[] incoming;
    Edge[] outgoing;
}

class Edge {
    Vertex start;
    Vertex end;
}

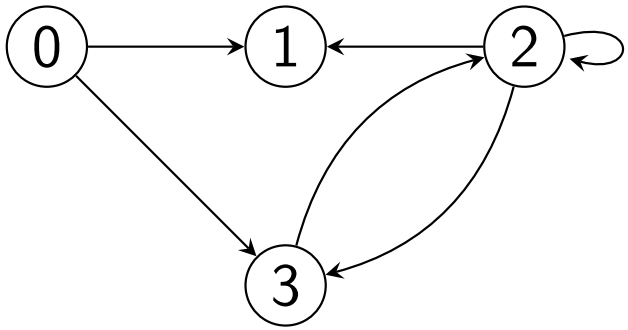
class Graph {
    Vertex[] vertices;
    Edge[] edges;
}
```

Adjazenzmatrix

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

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

- Beispiel:



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

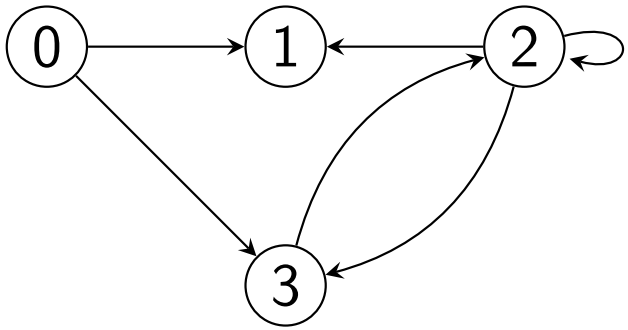
- Adjazenzmatrix eines ungerichteten Graphen $U = (V, E)$ ist die Adjazenzmatrix von $G = (V, E_g)$

Adjazenzmatrix

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

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

- Beispiel:



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

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

Wegematrix eines Graphen

- jede Relation als Matrix repräsentierbar
- z. B. auch die Erreichbarkeitsrelation E^*
- die zugehörigen Matrix heißt die *Wegematrix* W des Graphen,also

$$W[i,j] = \begin{cases} 1 & \text{falls } (i,j) \in E^* \\ 0 & \text{falls } (i,j) \notin E^* \end{cases}$$

$$= \begin{cases} 1 & \text{falls es in } G \text{ einen Pfad von } i \text{ nach } j \text{ gibt} \\ 0 & \text{falls es in } G \text{ keinen Pfad von } i \text{ nach } j \text{ gibt} \end{cases}$$

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

Was ist wichtig

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

Überblick

Die naheliegende Idee

- Benutze

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

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

Die naheliegende Idee

- Benutze

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

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

Die naheliegende Idee

- Benutze

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

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

Die naheliegende Idee

- Benutze

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

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

Beseitigung der *unendlichen* 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.
  - $p = (i_0, i_1, \ldots, i_k)$ 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

- 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.
  - $p = (i_0, i_1, \ldots, i_k)$ 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

- 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.
  - $p = (i_0, i_1, \ldots, i_k)$ 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

- 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.
  - $p = (i_0, i_1, \ldots, i_k)$ 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

- 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.
  - $p = (i_0, i_1, \ldots, i_k)$ 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:

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

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:

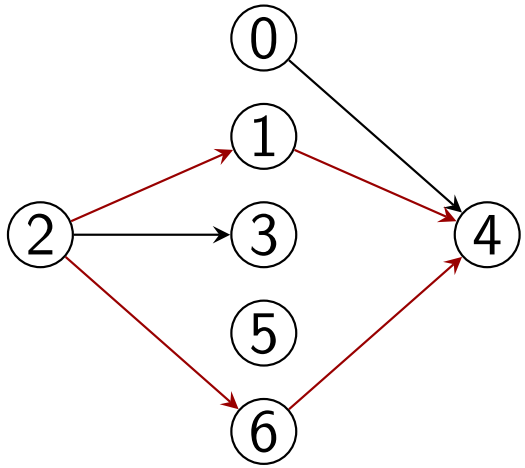
$$\forall k \geq n-1 : E^* = \bigcup_{i=0}^{k} E^i$$

Quadrierte Adjazenzmatrix: Beispiel

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

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

Quadrierte Adjazenzmatrix: Beispiel



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

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

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

Quadrierte Adjazenzmatrix

- 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.

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 $k = 2$.

Die Signum-Funktion

- *Signum-Funktion*

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

- Erweiterung auf Matrizen durch komponentenweise Anwendung

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

Matrizen für die Relationen E^k

Korollar

Es sei G ein gerichteter Graph mit Adjazenzmatrix A.
Für alle $k \in \mathbb{N}_0$ gilt:

1.

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

2. Matrix $\operatorname{sgn}(A^k)$ repräsentiert die Relation E^k.
   (A^0 sei die Einheitsmatrix.)

Formel für die Wegematrix

Lemma

Es sei G ein gerichteter Graph mit Adjazenzmatrix A.
Dann gilt für alle $k \geq n - 1$:

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

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

ist die Wegematrix des Graphen G.

Beweis

Man muss sich noch überlegen:

- $\bigcup_{i=0}^{n-1} E^i$ wird durch Matrix $\operatorname{sgn}(\sum_{i=0}^{k} \operatorname{sgn}(A^i))$ repräsentiert.
- In dieser Formel darf man die „inneren“ Anwendungen von sgn weglassen.

Das sieht man so:

- allgemein: Relationen R und R', durch Matrizen M und M' dargestellt. Dann

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

- Wenn alle Matrixeinträge ≥ 0 sind, gilt:

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

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 ← 0                          // Nullmatrix
for i ← 0 to n − 1 do
    M ← Id                     // Einheitsmatrix
    for j ← 1 to i do
        M ← M · A              // Matrixmultiplikation
    od
    W ← W + M                  // Matrixaddition
od
W ← sgn(W)
```

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**
    $S[i,j] \leftarrow M_1[i,j] + M_2[i,j]$
  od
od

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

Wieviele elementare Operationen für Matrixmultiplikation?

- Matrixmultiplikation

for $i \leftarrow 0$ **to** $m_1 - 1$ **do**
  for $j \leftarrow 0$ **to** $n_2 - 1$ **do**
    $P[i,j] \leftarrow 0$
    for $k \leftarrow 0$ **to** $n_1 - 1$ **do**
      $P[i,j] \leftarrow P[i,j] + M_1[i,k] \cdot M_2[k,j]$
    od
  od
od

- $m_1 \cdot n_2 \cdot n_1$ Additionen und $m_1 \cdot n_2 \cdot n_1$ Multiplikationen
- kleine Variante: $m_1 \cdot n_2 \cdot (n_1 - 1)$ 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:

$W \leftarrow 0$
for $i \leftarrow 0$ **to** $n-1$ **do**
  $M \leftarrow \mathrm{Id}$
  for $j \leftarrow 1$ **to** i **do**
    $M \leftarrow M \cdot A$
  od
  $W \leftarrow W + M$
od
$W \leftarrow \mathrm{sgn}(W)$

- 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:

$W \leftarrow 0$
for $i \leftarrow 0$ **to** $n-1$ **do**
  $M \leftarrow \mathrm{Id}$
  for $j \leftarrow 1$ **to** i **do**
    $M \leftarrow M \cdot A$
  od
  $W \leftarrow W + M$
od
$W \leftarrow \mathrm{sgn}(W)$

- 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:

$W \leftarrow 0$
for $i \leftarrow 0$ **to** $n-1$ **do**
  $M \leftarrow \mathrm{Id}$
  for $j \leftarrow 1$ **to** i **do**
    $M \leftarrow M \cdot A$
  od
  $W \leftarrow W + M$
od
$W \leftarrow \mathrm{sgn}(W)$

- 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:

$W \leftarrow 0$
for $i \leftarrow 0$ **to** $n-1$ **do**
  $M \leftarrow \mathrm{Id}$
  for $j \leftarrow 1$ **to** i **do**
    $M \leftarrow M \cdot A$
  od
  $W \leftarrow W + M$
od
$W \leftarrow \mathrm{sgn}(W)$

- 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:

$W \leftarrow 0$
for $i \leftarrow 0$ **to** $n-1$ **do**
  $M \leftarrow \mathrm{Id}$
  for $j \leftarrow 1$ **to** i **do**
    $M \leftarrow M \cdot A$
  od
  $W \leftarrow W + M$
od
$W \leftarrow \mathrm{sgn}(W)$

- 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:

$W \leftarrow 0$
for $i \leftarrow 0$ **to** $n-1$ **do**
  $M \leftarrow \mathrm{Id}$
  for $j \leftarrow 1$ **to** i **do**
    $M \leftarrow M \cdot A$
  od
  $W \leftarrow W + M$
od
$W \leftarrow \mathrm{sgn}(W)$

- Aufwand:

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

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:

  $W \leftarrow 0$
  $M \leftarrow \mathrm{Id}$
  for $i \leftarrow 0$ **to** $n-1$ **do**
    $W \leftarrow W + M$
    $M \leftarrow M \cdot A$
  od
  $W \leftarrow \mathrm{sgn}(W)$

- Aufwand:

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

Es geht noch besser!

- 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 $W = F^{2^m} = (\cdots((F^2)^2)\cdots)^2$
  - Das sind nur noch $m = \lceil \log_2 n \rceil$ Matrixmultiplikationen!
- Preisfrage: Wie sieht F aus?
- Antwort: Wähle $F = E^0 \cup E^1 = \mathrm{Id}_V \cup E$.

Es geht noch besser!

- 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 $W = F^{2^m} = (\cdots((F^2)^2)\cdots)^2$
  - Das sind nur noch $m = \lceil \log_2 n \rceil$ Matrixmultiplikationen!
- Preisfrage: Wie sieht F aus?
- Antwort: Wähle $F = E^0 \cup E^1 = \mathrm{Id}_V \cup E$.

Es geht noch besser!

- 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 $W = F^{2^m} = (\cdots((F^2)^2)\cdots)^2$
  - Das sind nur noch $m = \lceil \log_2 n \rceil$ Matrixmultiplikationen!
- Preisfrage: Wie sieht F aus?
- Antwort: Wähle $F = E^0 \cup E^1 = \mathrm{Id}_V \cup E$.

Es geht noch besser!

- 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 $W = F^{2^m} = (\cdots((F^2)^2)\cdots)^2$
  - Das sind nur noch $m = \lceil \log_2 n \rceil$ Matrixmultiplikationen!
- Preisfrage: Wie sieht F aus?
- Antwort: Wähle $F = E^0 \cup E^1 = \mathrm{Id}_V \cup E$.

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

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

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

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

Es geht noch besser! (3)

- Algorithmus:

$W \leftarrow A + \mathrm{Id}$
$m \leftarrow \lceil \log_2 n \rceil$
for $i \leftarrow 1$ **to** m **do**
  $W \leftarrow W \cdot W$
od
$W \leftarrow \mathrm{sgn}(W)$

- 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.

Was ist wichtig

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