

12 ERSTE ALGORITHMEN IN GRAPHEN

12.1 REPRÄSENTATION VON GRAPHEN IM RECHNER

Man vergleiche Adjazenzliste und Adjazenzmatrizen:

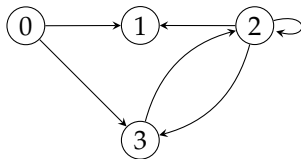
- Bei den Listen hat man „schnell“ Zugriff auf alle adjazenten Knoten, bei der Matrix muss man alle Knoten überhaupt durchgehen, um zu sehen, welche Nachbarn sind.
- Bei der Matrix kann man „schnell“ herausfinden, ob es eine Kante zwischen zwei Knoten i und j gibt, bei den Listen muss man unter Umständen alle Nachbarn durchgehen.
- Wann spart was Speicherplatz? (Listen bei „relativ wenigen“ Knoten)

Adjazenzmatrizen:

- Woran erkennt man eine Schleife? (1 auf der Diagonale)
- Beispiele machen, z. B. $A =$ alles Einsen
- welche besondere Eigenschaft haben die Adjazenzmatrizen ungerichteter Graphen? (Symmetrie Hauptdiagonale)
- Beispiele von Matrix zum Graphen und zurück

Wegematrix:

- erst mal einfach durch Hingucken für einen Graphen eine bestimmen, z. B. für



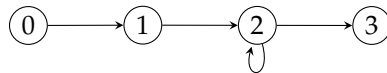
$$W = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} & \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} \end{matrix}$$

- Wie sieht die Wegematrix aus, wenn $A =$ alles Einsen? ($W = A$)
- etwas schwieriger: Wann ist allgemein $W = A$? (Wenn Kantenrelation reflexiv und transitiv)

12.2 BERECHNUNG DER 2-ERREICHBARKEITSRELATION UND RECHNEN MIT MATRIZEN

Berechnung von E^2

- wie im Skript noch ein ausführliches Beispiel, vielleicht für



12.2.1 Matrixmultiplikation

Matrixmultiplikation:

- zumindest die Infowirte müssen das üben. Man nehme die 4×4 -Matrizen mit $A_{ij} = \begin{cases} 1 & \text{falls } i \leq j \\ 0 & \text{sonst} \end{cases}$ und $B_{ij} = \begin{cases} 1 & \text{falls } i > j \\ 0 & \text{sonst} \end{cases}$ und berechne AB und BA

- aus der Vorlesung; noch mal durchgehen:

```

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

```

Einheitsmatrizen

- $I \cdot A = A$ nachrechnen:

$$(I \cdot A)_{ij} = \sum_{v=0}^{n-1} I_{iv} \cdot A_{vj} = I_{ii} \cdot A_{ij} = A_{ij}$$

12.2.2 Matrixaddition

12.3 BERECHNUNG DER ERREICHBARKEITSRELATION

12.3.1 Potenzen der Adjazenzmatrix

quadrierte Adjazenzmatrix:

- inhaltliche Bedeutung von $(A^2)_{ij}$: unbedingt klar machen

12.3.2 Erste Möglichkeit für die Berechnung der Wegematrix

12.3.3 Zählen durchzuführender arithmetischer Operationen

Zählen von Operationen:

- so was wie $\sum_{i=1}^{n-1} i = n(n-1)/2$ klar machen
- ich benutze immer die Methode, mit der Gauß schon als Schulkind auffiel, als alle Schüler $1 + 2 + 3 + \dots + 100$ ausrechnen sollten: erster + letzter Summand = zweiter + vorletzter Summand = ..., also insgesamt Wert = so eine Zweiersumme mal Anzahl Summanden halbe (funktioniert auch bei ungerader Anzahl Summanden)

12.3.4 Weitere Möglichkeiten für die Berechnung der Wegematrix

Wegematrix schneller (n^4):

- Wenn man eine feste Gesamtzeit zur Verfügung hat und mit dem n^5 Algorithmus gerade noch Probleminstanzen mit $n = 1000$ schafft: Wie große Probleminstanzen schafft man mit dem n^4 Algorithmus?

Von der Rechenregel

$$(A \cup B) \circ (C \cup D) = (A \circ C) \cup (A \circ D) \cup (B \circ C) \cup (B \circ D)$$

für Relationen kann man sich mal einen Teil klar machen, z. B. falls alle binären Relationen auf Menge M sind:

$$(A \cup B) \circ C = (A \circ C) \cup (B \circ C)$$

(anderer Teil analog) Beweis durch Nachprüfen beider Inklusionen, evtl. gleichzeitig:

$$\begin{aligned}(x, z) \in (A \cup B) \circ C &\iff \exists y \in M : (x, y) \in A \cup B \wedge (y, z) \in C \\ &\iff \exists y \in M : ((x, y) \in A \vee (x, y) \in B) \wedge (y, z) \in C \\ &\iff \exists y \in M : ((x, y) \in A \wedge (y, z) \in C) \vee ((x, y) \in B \wedge (y, z) \in C) \\ &\iff \exists y \in M : ((x, y) \in A \wedge (y, z) \in C) \\ &\quad \vee \exists y \in M : ((x, y) \in B \wedge (y, z) \in C) \\ &\iff (x, z) \in A \circ C \vee (x, z) \in B \circ C \\ &\iff (x, z) \in A \circ C \cup B \circ C\end{aligned}$$

Allerdings:

- beim dritten \iff braucht man Distributivgesetz für Aussagenlogik: durch Nachdenken klar machen
- beim vierten \iff auch **unbedingt genau nachdenken**: es ist wichtig, dass da ein **Oder** steht
- für \wedge wäre das folgende **FALSCH FALSCH FALSCH** :

$$\exists y \in M : (\mathcal{A}(y) \wedge (y)) \iff \exists y \in M : \mathcal{A}(y) \wedge \exists y \in M : (y)$$

HIER IST \iff FALSCH: ES GILT NUR \implies !

- Gegenbeispiel $\exists y \in \mathbb{N}_0 : y = 1$ und $\exists y \in \mathbb{N}_0 : y = 2 \dots$
- Wenn Sie sich nicht in der Lage sehen, das klar rüber zu bringen, dann lieber im Beweis umgangssprachlich argumentieren.

Wegematrix schneller ($n^3 \log_2 n$):

- Wenn man eine feste Gesamtzeit zur Verfügung hat und mit dem n^5 Algorithmus gerade noch Probleminstanzen mit $n = 1000$ schafft: Wie große Probleminstanzen schafft man mit dem $n^3 \log_2 n$ Algorithmus?

12.4 ALGORITHMUS VON WARSHALL

Algorithmus von Warshall

- Korrektheit möglichst klar machen, denn der gleiche Trick wird vielleicht noch mal auftauchen bei der Konstruktion von regulären Ausdrücken zu endlichen Automaten.
- stumpfsinniges Nachvollziehen der Rechnereien des Algorithmus finde ich nur mäßig erhellend. Hat jemand noch schöne Bilder dazu?