16 ERSTE ALGORITHMEN IN GRAPHEN

16.1 Repräsentation von Graphen im Rechner

Man vergleiche Adjazenzliste und Adjazenzmatrizen:

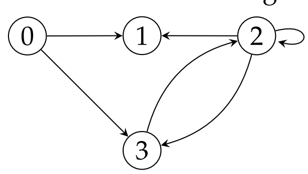
- Bei den Listen hat man „schnell" Zugriff auf alle adjazenten Knoten, bei der Matrix muss man alle Knoten überhaupt durchgehen, um zu sehen, welche Nachbarn sind.
- Bei der Matrix kann man „schnell" herausfinden, ob es eine Kante zwischen zwei Knoten i und j gibt, bei den Listen muss man unter Umständen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

Adjazenzmatrizen:

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

Wegematrix:

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



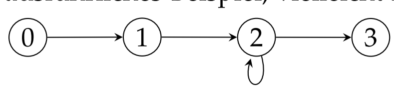
$$W = \begin{array}{c c} & \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} & \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} \end{array}$$

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

16.2 Berechnung der 2-Erreichbarkeitsrelation und Rechnen mit Matrizen

Berechnung von E^2

- wie im Skript noch ein ausführliches Beispiel, vielleicht für



16.2.1 Matrixmultiplikation

Matrixmultiplikation: sollten dieses Jahr *alle* schon gehabt haben. Aber Üben kann nicht schaden:

- Man nehme z. B. 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
- $\mathrm{I} \cdot A = A$ nachrechnen:

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

16.2.2 Matrixaddition

16.3 Berechnung der Erreichbarkeitsrelation

16.3.1 Potenzen der Adjazenzmatrix

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

16.3.2 Erste Möglichkeit für die Berechnung der Wegematrix

16.3.3 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 angeblich Gauß schon als Schulkind auffiel, als alle Schüler $1 + 2 + 3 + \cdots + 100$ ausrechnen sollten: erster + letzter Summand = zweiter + vorletzter Summand = . . . , also insgesamt Wert = so eine Zweiersumme mal Anzahl Summanden halbe (funktioniert auch bei ungerader Anzahl Summanden)

16.3.4 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 alles binäre Relationen auf Menge M sind:

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

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

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

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 $\land$ wäre das folgende **FALSCH FALSCH FALSCH** :

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

  HIER IST $\iff$ FALSCH: ES GILT NUR $\Longrightarrow$!
- 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?

16.4 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?

