Prof. Dr. Klaus Volbert Regensburg, den 16. Dezember 2025
Votieren und abgeben bis zum 22. Dezember 2025, 23:00 Uhr

Übungen zur Vorlesung

Algorithmen und Datenstrukturen

WiSe 2025/26

Blatt 11

Wichtige Hinweise:

- $->$ Falls Sie bei der Bearbeitung einer Aufgabe größere Schwierigkeiten hatten und deswegen die Bearbeitung abgebrochen haben, so versuchen Sie bitte Ihre Schwierigkeiten in Form von Fragen festzuhalten. Bringen Sie Ihre Fragen einfach zur Vorlesung oder zur Übung mit!
- $->$ Musterlösungen werden bei Bedarf in den Übungen besprochen!

Aufgabe 1:
Sei ein Graph $G = (V, E)$ durch folgende Adjazenzmatrix gegeben:

	0	1	2	3	4	5	6	7	8
0		1	1				1		
1				1					
2	1			1	1				
3							1		
4	1					1			1
5			1	1	1			1	
6								1	
7				1					1
8						1			

- Ist G ein gerichteter oder ungerichteter Graph?
- In welcher Reihenfolge werden die Knoten, beginnend mit Knoten 0, bei einer Breiten- und einer Tiefensuche besucht (sollten mehrere Knoten zur Wahl stehen, so wird immer der Knoten mit der kleinsten Nummer gewählt)? Implementieren Sie Breiten- und Tiefensuche und überprüfen Sie Ihre Ergebnisse anhand Ihrer Implementierung.
- Sei $A = (a_{ij})$ die Adjazenzmatrix eines Graphen $G = (V, E)$ mit Dimension $n \in \mathbb{N}$. Welche Bedeutung haben die Einträge der Matrix A^m für $m \in \mathbb{N}$?

Aufgabe 2:
Die *Inzidenzmatrix* eines gerichteten Graphen $G = (V, E)$ ohne Schlingen ist eine $|V| \times |E|$ Matrix $B = (b_{ij})$ mit:

$$b_{ij} = \begin{cases} -1, & \text{falls Kante } j \text{ Knoten } i \text{ verlässt (ausgehende Kante)} \\ 1, & \text{falls Kante } j \text{ zu Knoten } i \text{ führt (eingehende Kante)} \\ 0, & \text{sonst} \end{cases}$$

Geben Sie an, was die Einträge des Matrixproduktes BB^τ bedeuten, wenn B^τ die transponierte Matrix von B ist.

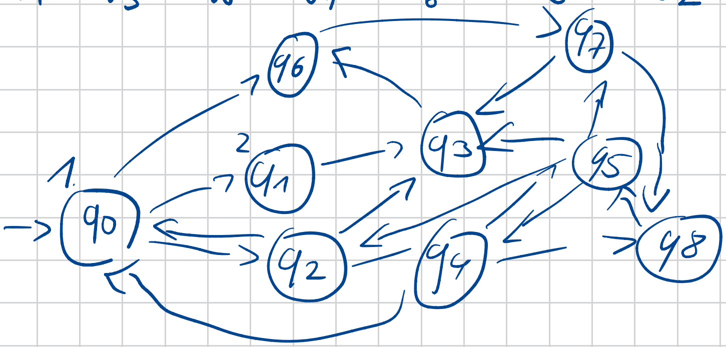
Aufgabe 1)

1.1) Da Adjazenzmatrix asymmetrisch => gerichteter Graph

1.2) Breitensuche: $V_0 \to V_1 \to V_2 \to V_6 \to V_3 \to V_4 \to V_7 \to V_5 \to V_8$

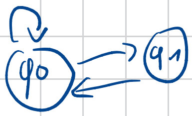
Tiefensuche: $V_0 \to V_1 \to V_3 \to V_6 \to V_7 \to V_8 \to V_5 \to V_2 \to V_4$

	0	1	2	3	4	5	6	7	8
0		1	1				1		
1				1					
2	1			1	1				
3							1		
4	1					1			1
5			1	1	1			1	
6								1	
7				1					1
8						1			



1.3) m = Pfadlänge, a_{ij} der Matrix A^m ist die Anzahl der Pfade von knoten i zu knoten j mit exakt m kanten.

Bsp.: Gegeben $A = \begin{pmatrix}1 & 1\\ 1 & 0\end{pmatrix}$,



, wir suchen wie viele Pfade es von q_0 zu q_0 mit Pfadlänge 2 gibt.

$q_0 \to q_0 = a_{11}$

$A^2 = \begin{pmatrix}2 & 1\\ 1 & 1\end{pmatrix}$ => 2 Pfade gibt es: $q_0 \to q_0 \to q_0$ ✓ und $q_0 \to q_1 \to q_0$ ✓

Aufgabe 2)

Sei G ein gerichteter Graph mit $G = (V,E)$, $|V| \times |E|$ und ohne Schlingen

Matrix $B = (b_{ij})$ mit:

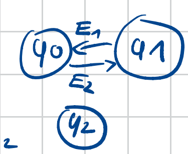
$$b_{ij} = \begin{cases} -1, & \text{falls kante } j \text{, knoten } i \text{ verlässt (ausgehende kante)} \\ 1, & \text{falls kante } j \text{ auf knoten } i \text{ führt (eingehende kante)} \\ 0, & \text{sonst} \end{cases}$$

$B = |V| \times |E|$, $B^T - |E| \times |V|$ => $BB^T = |V| \times |V|$

Bsp.: $B = \begin{pmatrix}1 & -1\\ -1 & 1\\ 0 & 0\end{pmatrix}$,

V/E	E_1	E_2
q_0	1	-1
q_1	-1	1
q_2	0	0

,



$BB^T = \begin{pmatrix}1 & -1\\ -1 & 1\\ 0 & 0\end{pmatrix} \cdot \begin{pmatrix}1 & -1 & 0\\ -1 & 1 & 0\end{pmatrix} = \begin{pmatrix}2 & -2 & 0\\ -2 & 2 & 0\\ 0 & 0 & 0\end{pmatrix}$,

V/V	q_0	q_1	q_2
q_0	2	-2	0
q_1	-2	2	0
q_2	0	0	0

=> { Hauptdiagonale, # Aller kanten a_{ij}, wobei $i = j$
Nebendiagonale, # der gemeinsamen kanten zwisch i und j
↳ { a_{ij} (-1) wenn existiert; 0, sonst

Aufgabe 3:
Entwerfen und implementieren Sie einen Algorithmus, der bei Eingabe eines gerichteten Graphen $G = (V, E)$ einen Zyklus ausgibt, falls einer vorhanden ist.

gerichtet

Aufgabe 4:
Beschreiben Sie die Implementierung eines Algorithmus, der für einen Dag $G = (V, E)$ eine topologische Sortierung in Zeit $O(|V| + |E|)$ berechnet, indem er in G wiederholt nach einem Knoten mit Eingangsgrad 0 sucht und diesen aus G inkl. seiner ausgehenden Kanten entfernt. Wie reagiert der Algorithmus auf Zyklen?

→ von U nach V ⇒ U steht vor V

Aufgabe 3) => mit Tiefensuche als Ausgabe

Sei $G = (V, E)$ ein gerichteter Graph und die Zustände.

- white = Knoten wurde noch nicht besucht
- grey = Knoten liegt im Pfad der Tiefensuche
- black = Knoten ist abgearbeitet

- Zyklus (G)
  - Setze jeden Knoten:
    - Farbe = white
    - Vorgänger = null

  - Für jeden Knoten v im Graphen G:
    - Falls v noch unbesucht:
      - Starte Tiefensuche von v
      - Falls Zyklus gefunden (rufe Tiefensuche Funktion auf):
        - Beende Algo

  - Falls kein Zyklus gefunden:
    - "kein Zyklus"

- Tiefensuche(c)
  - c wird grey (Teil des Pfades)
  - Für jeden Nachfolger v von c
    - Falls v white:
      - Setze Vorgänger v auf c
      - Führe Tiefensuche (v) aus

    - Sonst falls v grey:
      - "Zyklus wurde gefunden"
      - Rekonstruiere Zyklus über Vorgänger (Print Funktion)
      - Gibt Zyklus aus
      - Beende Algo

  - c wird black (entfernt c aus aktuellem Pfad)

Aufgabe 4)

- Berechne für jeden Knoten den Eingangsgrad
- Alle Knoten mit Grad 0 in eine queue
- Schleife so lange bis keine Knoten:
  - entferne so einen Knoten
  - füge ihn zur Ausgabe (also der topologische Sortierung) hinzu
  - entferne seine ausgehenden Kanten und aktualisiere den Grad

Was passiert bei Zyklen?

Hat der Graph noch existierende Knoten die nicht Eingangsgrad 0 haben => Zyklus (keine topol. Sortierung möglich)