Algorithmische Mathematik:
Graphen & Anwendungen

Frühlingssemester 2018
P. Zaspel und I. Kalmykov

Universität Basel

Übungsblatt 2.

zu bearbeiten bis **Dienstag, 13.3.2018, 14:00 Uhr**.

Aufgabe 1. (Graphen und Adjazenzmatrizen)

Wir betrachten den folgenden gerichteten Graphen in der Abbildung 1, in dem $\leftrightarrow$ für eine Doppelkante steht (d.h. Kurzform für je eine Kante *vom* Knoten und eine Kante *zum* Knoten).

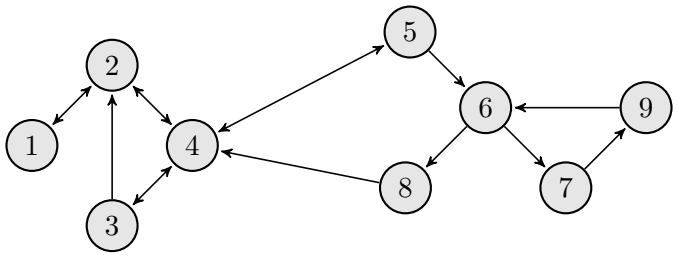

Abbildung 1: Graph Aufgabe 1

a) Geben Sie alle einfachen Wege vom Knoten 1 zum Knoten 8 sowie vom Knoten 8 zum Knoten 1 an.

b) Geben Sie die Adjazenzmatrix für obigen Graphen an. Geben Sie auch die Adjazenzmatrix an, wenn alle Kanten *ungerichtet* wären.

c) Zeichnen Sie einen Graphen mit Knoten $1, 2, \ldots, 9$, der die folgende Adjazenzmatrix besitzt.

	1	2	3	4	5	6	7	8	9
1		1	1		1				
2	1		1				1		
3	1	1							
4					1				
5	1			1					
6	1						1		
7		1				1		1	1
8							1		
9				1			1		

Muss der Graph gerichtet gezeichnet werden oder kann er auch ungerichtet sein?

(4 Punkte)

Aufgabe 2. (Lemma 1.3 aus der Vorlesung)

a) Sei $G = (V, E)$ ungerichtet, $v \in V(G)$. Zeigen Sie

$$\sum_{v \in V(G)} |\text{post}(v)| = \sum_{v \in V(G)} |\text{pre}(v)| = 2\,|E|\,.$$

b) Sei $H = (V, E)$ gerichtet, $v \in V(H)$. Zeigen Sie

$$\sum_{v \in V(H)} |\text{post}(v)| = \sum_{v \in V(H)} |\text{pre}(v)| = |E|\,.$$

c) Folgern Sie, dass in jedem Graphen eine gerade Anzahl von Knoten mit ungeradem Grad gibt.

(4 Punkte)

Aufgabe 3. (Anzahl Kanten)

Sei $G = (V, E)$ ein ungerichteter Graph mit $n = |V| \geq 3$ Knoten. Zeigen Sie, dass dann folgende Aussagen gelten:

a)

$$|E| = \frac{1}{n-2} \sum_{v \in V} |E'(v)|.$$

Dabei sei $G'(v) = (V'(v), E'(v))$ derjenige Teilgraph, der aus G durch Streichung von v sowie aller an v anliegenden Kanten entsteht.

b) Falls nun $n > 3$ und $|E| > \frac{n^2}{4}$ gilt, so gibt es einen Knoten $v \in V$, so dass $G'(v)$ mehr als $\frac{(n-1)^2}{4}$ Kanten hat.

(4 Punkte)

Aufgabe 4. (Implementierung von Graphen)

Für die Implementierung von Graphenalgorithmen beschränken wir uns im Folgenden auf Graphen $G = (V, E)$ mit $V = \{1, \ldots, n\}$, $n \in \mathbb{N}_{>0}$.

a) Eine Möglichkeit, Graphen abzuspeichern sind Adjazenzlisten. Dabei wird für jeden Knoten $v \in V(G)$ eine Liste mit allen $w \in \text{post}(v)$ erstellt. Im Folgenden bezeichnet $\mathbf{A}$ die Menge der Adjazenzlisten zu einem Graphen, $\mathbf{A}_i$ die Adjazenzliste zu dem Knoten i und $\mathbf{A}_{i,j}$ das j-te Element der i-ten Adjazenzliste.

Implementieren Sie eine Datenstruktur zum speichern von $\mathbf{A}$. In Matlab können Sie hierzu Cell Arrays benutzen, wobei die Zelle mit dem Index i die Adjazenzliste zu dem Knoten mit der Nummer i enthält. Die jeweiligen Adjazenzlisten können als numerische Arrays abgespeichert werden. Basierend auf Ihrer Implementierung der Adjazenzlisten entwickeln Sie Funktionen für das Einfügen (Algorithmus 1) und Entfernen von Kanten (Algorithmus 2). Die implementierten Funktionen sollen auf den Graphen in Abbildung 2 angewendet werden. Testen Sie ihre Implementierung an folgenden Aufgaben:

(i) Implementieren und testen Sie eine Funktion zum Ausgeben von $\mathbf{A}$.

(ii) Entfernen Sie die Kanten $\{2, 3\}$ und $\{4, 5\}$ und geben sie das modifizierte $\tilde{\mathbf{A}}$ aus.

(iii) Wie sieht die $\tilde{\mathbf{A}}$ aus, wenn danach noch die Kante $\{1, 5\}$ hinzufügt wird?

Bemerkung. Der Graph in der Abbildung 2 ist ungerichtet. Wenn man also eine Kante $\{u, v\}$ hinzufügen oder entfernen möchte, muss man entsprechend die gerichteten Kanten (u, v) und (v, u) hinzufügen oder entfernen.

b) Wir möchten eine Möglichkeit haben, Algorithmen auf Graphen effizient zu debuggen. Implementieren Sie dazu eine Routine um einen Graph analog zu den Abbildungen 1 bzw. 2 zu plotten. Dabei sollen die Positionen der Knoten gemäß dem Hall's spektralen Ansatz bestimmt werden. Testen Sie Ihre Implementierung an dem Graphen aus Abbildung 2.

(4 Punkte)

Algorithmus 1 addEdge $(\mathbf{A}, u, v)$ - Hinzufügen der Kante (u, v)

Input: Menge der Adjazenzlisten $\mathbf{A}$, Knoten u und Knoten v der neuen Kante
Output: modifizierte Menge der Adjazenzlisten $\tilde{\mathbf{A}}$

1: $\tilde{\mathbf{A}} \leftarrow \mathbf{A}$
2: **if** $\forall i \in \{1, \ldots, |\mathbf{A}_u|\} : \mathbf{A}_{u,i} \neq v$ **then**
3: $\quad \tilde{\mathbf{A}}_{u,|\mathbf{A}_u|+1} \leftarrow v$
4: **end if**
5: **return** $\tilde{\mathbf{A}}$

Algorithmus 2 removeEdge $(\mathbf{A}, u, v)$ - Entfernen der Kante (u, v)

Input: Menge der Adjazenzlisten $\mathbf{A}$, Knoten u und Knoten v der zu entferndenden Kante
Output: modifizierte Menge der Adjazenzlisten $\tilde{\mathbf{A}}$

1: $\tilde{\mathbf{A}} \leftarrow \mathbf{A}$
2: **if** $\exists i \in \{1, \ldots, |\mathbf{A}_u|\} : \mathbf{A}_{u,i} = v$ **then**
3: $\quad \tilde{\mathbf{A}}_u \leftarrow \mathbf{A}_u \setminus v$
4: **end if**
5: **return** $\tilde{\mathbf{A}}$

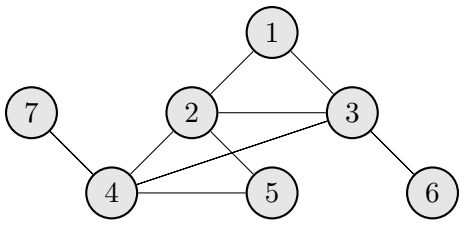

Abbildung 2: Graph Aufgabe 4