ETH Zürich
Institut für Theoretische Informatik
Prof. Dr. Angelika Steger, Dr. Johannes Lengler
Übungsleitung: Florian Meier

HS 2019
Abgabe 01.10.2019

Algorithmen und Komplexität
Übungsblatt 2

Aufgabe 1 wird von Ihren Mitstudierenden korrigiert. Die Partnerzulosung findet in der Übungsstunde statt. Am Dienstag tauschen Sie die Ihre Lösung dann direkt mit Ihrem*Ihrer Partner*in zur Korrektur aus. Die Rückgabe und das Austauschen des Feedbacks sollte in derselbe Woche erfolgen. Aufgaben 2 und 3 werden von den Assistenten korrigiert.

⋆ ⋆ ⋆

Aufgabe 1

(a) Der Graph $G = (V, E)$ habe n Knoten und m Kanten; für einen Knoten $v \in V$ bezeichne $\deg(v)$ seinen Grad. Bestimmen Sie die Anzahl der Kanten sowie den Grad der Knoten im Komplement $\overline{G}$ von G.

(b) Der d-dimensionale Hyperwürfel Q_d ist wie folgt definiert: Die Knotenmenge ist die Menge aller 0-1-Folgen der Länge d, und zwei Knoten sind jeweils verbunden, wenn sich die entsprechenden Folgen in genau einer Stelle unterscheiden. Bestimmen Sie die Anzahl der Knoten und der Kanten in Q_d. Was ist die Länge des kürzesten Pfades zwischen zwei Knoten $u, v \in \{0, 1\}^d$?

(c) Zeigen Sie, dass jeder Graph auf mindestens zwei Knoten mindestens zwei Knoten besitzt, die denselben Grad haben.

(d) Zeigen Sie, dass in jedem Graphen die Anzahl der Knoten mit ungeradem Grad gerade ist.

Aufgabe 2

Aus der Vorlesung wissen wir (Satz 1.18): Ist $G = (V, E)$ ein zusammenhängender Graph mit $|E| = |V| - 1$, so ist G kreisfrei und somit ein Baum. Zeigen Sie:

(a) Ist G zusammenhängend und kreisfrei, so gilt $|E| = |V| - 1$.

(b) Ist G kreisfrei und gilt $|E| = |V| - 1$, so ist G zusammenhängend.

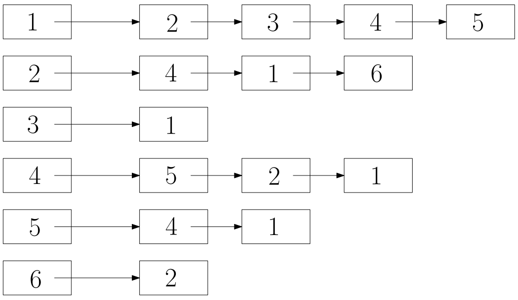
Aufgabe 3

In dieser Aufgabe betrachten wir drei Möglichkeiten Graphen abzuspeichern und analysieren wie effizient damit grundlegende Graphenoperationen ausgeführt werden können.

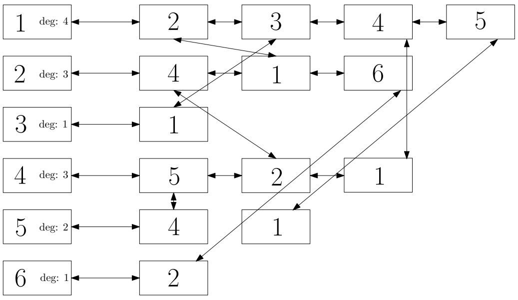
1) **Einfache Adjazenzmatrix:**
Der Graph wird in einer $n \times n$-Matrix abgespeichert. An jeder Stelle der Matrix steht eine Eins, falls im Graph eine Kante zwischen den entsprechenden Knoten ist, und eine Null sonst.

$$\text{Adjazenz Matrix } A_G\text{:}\ \ (A_G)_{i,j} = \begin{cases} 1, & if\ \{i,j\} \in E \\ 0, & if\ \{i,j\} \notin E \end{cases}$$

2) **Einfache Adjazenzliste**
Der Graph wird durch n Listen gespeichert. Jede Liste beschreibt die Nachbarn eines Knotens. Die Listen sind nur "vorwärts verlinkt", das heisst, vom k-ten Nachbarn eines Knotens kann direkt auf den $k+1$-ten Nachbarn zugegriffen werden, während die Liste von vorne durchlaufen werden muss um auf den $k-1$-ten Nachbarn zuzugreifen.



3) **Erweiterte Adjazenzliste:**
Der Graph wird in n Listen gespeichert. Jede Liste beschreibt wiederum die Nachbarn eines Knotens. Am Anfang jeder Liste wird zusätzlich der Grad des Knotens gespeichert. Desweitern sind die Listen vorwärts und rückwärts verlinkt und jeder Eintrag besitzt einen Pointer zum korrespondierenden Eintrag in der Liste des benachbarten Knoten.



Was ist jeweils der Speicherverbrauch (in O-Notation)? Welche Laufzeit (worst case, in O-Notation) bekommt man jeweils für die folgenden Probleme?

(i) Input: Ein Knoten $v \in V$. Finde $\deg(v)$.

(ii) Input: Ein Knoten $v \in V$. Gebe einen beliebigen Nachbarn von v aus.

(iii) Input Zwei Knoten $u, v \in V$. Sind u und v benachbart?

(iv) Input: Eine Kante $\{u, v\} \in E$. Lösche e aus dem Graphen.

(v) Input: $u, v \in V$ mit $u \neq v$. Füge eine Kante $\{u, v\}$ in den Graphen ein.

(vi) Input: $v \in V$. Lösche v und alle inzidenten Kanten aus dem Graphen.

ABGABE DER HAUSAUFGABEN IN DER VORLESUNG AM 01.10.2019.