Berlin, 1. August 2019

Name: .... **Matr.-Nr.: ....**

Klausur Grundlagen der Algorithmik

(Niedermeier/Bentert, Sommersemester 2019)

Aufgabe Nr.:	1	2	3	4	5	**Summe**
Punktzahl:	10	6	15	9	10	50
Davon erreicht:						

Einlesezeit:	15 Minuten
Bearbeitungszeit:	60 Minuten
Max. Punktezahl:	50 Punkte

Allgemeine Hinweise:

- Es sind keinerlei Hilfsmittel erlaubt.
- Benutzen Sie keinen Bleistift, sondern einen Kugelschreiber in der Farbe Schwarz oder Blau.
- Beschriften Sie jedes Blatt mit Vor- und Nachnamen sowie Matrikelnummer.
- **Falls in der Aufgabenstellung nicht explizit ausgeschlossen, so sind alle Antworten zu begründen! Antworten ohne Begründung erhalten 0 Punkte.**
- Sätze aus der Vorlesung dürfen ohne Beweis verwendet werden.

Viel Erfolg!

Aufgabe 1: **Modellieren mit Maximum Flow** (10 Punkte)

Ein Stadtteil soll teilweise neu bebaut werden. Dazu wurde der Stadtteil in $3n^2$ Gebiete unterteilt, die in $3n$ senkrechten und n waagerechten Reihen angeordnet sind. In Abbildung 1 ist ein Beispiel mit $n = 2$ gegeben. Nun sollen $3n$ Hochhäuser und einige niedrigere Gebäude gebaut werden. Aus logistischen Gründen soll

- in jedem Gebiet maximal *ein Hochhaus* gebaut werden, und
- es sollen in jeder waagerechten Reihe *drei* und
- in jeder senkrechten Reihe *ein* Hochhaus gebaut werden.

Ihnen ist bekannt, wie viele neue Gebäude in jedem Gebiet gebaut werden sollen, und Ihre Aufgabe besteht darin, zu entscheiden, ob es möglich ist, die zu bauenden Hochhäuser so auf die Gebiete aufzuteilen, dass alle Anforderungen erfüllt werden. Es kann davon ausgegangen werden, dass jedes geplante Gebäude auf jedem verfügbaren Platz gebaut werden kann.

Modellieren Sie dieses Problem als ein MAXIMUM FLOW-Problem, sodass die Anzahl der Knoten und die Anzahl der Kanten im Flussnetzwerk polynomiell in n sind. Geben Sie hierfür die Knoten, Kanten und Kantenkapazitäten Ihres Flussnetzwerks an und beschreiben Sie, wie die Antwort bestimmt werden kann.

Hinweis: Es kann auch Gebiete geben, in denen keine neuen Gebäude gebaut werden sollen.

0	0	2	2	3	1
4	1	2	0	3	0

Abbildung 1: Ein Beispiel mit $n = 2$. Jedes Quadrat illustriert ein Gebiet und die Zahl darin beschreibt, wie viele Gebäude in diesem Gebiet neu gebaut werden sollen. Eine mögliche Lösung in diesem Beispiel besteht darin, in den drei linken unteren und in den drei rechten oberen Gebieten jeweils ein Hochhaus zu bauen.

Aufgabe 2: **Eigenschaften von Netzwerkflüssen** (6 Punkte)

In dieser Aufgabe soll der maximale s-t-Fluss in einem einfachen, gerichteten Graph diskutiert werden.

Wie groß kann der maximale s-t-Fluss in einem Flussnetzwerk aus $n \geq 3$ Knoten höchstens sein, wenn jede Kante Kapazität 1 hat? Begründen Sie, warum es keinen Graph gibt, in dem der s-t-Fluss größer ist und geben Sie einen Graph an, in dem der s-t-Fluss genau Ihrer Antwort entspricht.

Erinnerung: Ein gerichteter Graph ist einfach, wenn er keine Schlaufen (Kanten der Form (v, v)) enthält und keine Kante mehrfach vorkommt.

Aufgabe 3: **Modellieren mit ILP** (15 Punkte)

Betrachten Sie das folgende Problem:

Bandwidth
Eingabe: Ein ungerichteter Graph $G = (V, E)$.
Aufgabe: Weise jedem Knoten $v \in V$ eine ganze Zahl $f(v)$ zwischen 1 und $n := |V|$ zu, sodass keine zwei Knoten den selben Wert zugewiesen bekommen und $\max\{|f(v) - f(w)| \mid \{v, w\} \in E\}$ (die „Länge der längsten Kante") minimiert wird.

(a) Zeigen Sie, dass die folgende ILP-Formulierung für Bandwidth korrekt ist. Geben Sie dazu an, wofür die Variablen und die Zielfunktion stehen, was die einzelnen Nebenbedingungen aussagen und wie das Ergebnis aus einer ILP-Lösung abgelesen werden kann. (10 P)

$$\begin{aligned}
\text{Variablen: } & f_v && \forall v \in V \\
& s_{v,w} && \forall v, w \in V \\
& k && \\
\text{Zielfunktion: } & \text{Minimiere } k &&
\end{aligned}$$

$$\begin{aligned}
\text{Nebenbedingungen: } & 1 \leq f_v \leq n && \forall v \in V && (1) \\
& f_v \in \mathbb{N} && \forall v \in V && (2) \\
& s_{v,w} \in \{0, 1\} && \forall v, w \in V (v \neq w) && (3) \\
& f_v - f_w \geq 1 - n \cdot s_{v,w} && \forall v, w \in V (v \neq w) && (4) \\
& f_w - f_v \geq 1 - n \cdot (1 - s_{v,w}) && \forall v, w \in V (v \neq w) && (5) \\
& f_v - f_w \leq k && \forall \{v, w\} \in E && (6) \\
& f_w - f_v \leq k && \forall \{v, w\} \in E && (7)
\end{aligned}$$

(b) Wandeln Sie die obige ILP-Formulierung ab, um das folgende Problem Tapewidth zu lösen. Beschreiben Sie nur, was sich ändert und wie daraus die Korrektheit der neuen Formulierung für Tapewidth folgt. (5 P)

Tapewidth
Eingabe: Ein ungerichteter Graph $G = (V, E)$.
Aufgabe: Weise jedem Knoten $v \in V$ eine ganze Zahl $f(v)$ zwischen 1 und $n := |V|$ zu, sodass keine zwei Knoten den selben Wert zugewiesen bekommen und $\sum_{\{v,w\} \in E} |f(v) - f(w)|$ minimiert wird.

Aufgabe 4: **Parametrisierte Algorithmen** (9 Punkte)

Geben Sie einen Algorithmus an, der CLIQUE parametrisiert durch ein „Feedback Vertex Set“ F in $2^{|F|} \cdot \text{poly}(n+m)$ Schritten löst, wobei n und m die Anzahl der Knoten und Kanten im Eingabegraph sind. Das Feedback Vertex Set F ist Ihnen als Eingabe gegeben und ist eine Menge von Knoten, sodass der Eingabegraph ein Wald wird, wenn die Knoten aus F und alle daran anliegenden Kanten aus dem Eingabegraph gelöscht werden. Begründen Sie, warum Ihr Algorithmus korrekt ist und warum er nach spätestens $2^{|F|} \cdot \text{poly}(n+m)$ Schritten terminiert.

Hinweis: Sie können davon ausgehen, dass Sie für zwei beliebige Knoten u, v in konstanter Zeit überprüfen können, ob die Kante $\{u, v\}$ im Eingabegraph existiert.

Erinnerung:

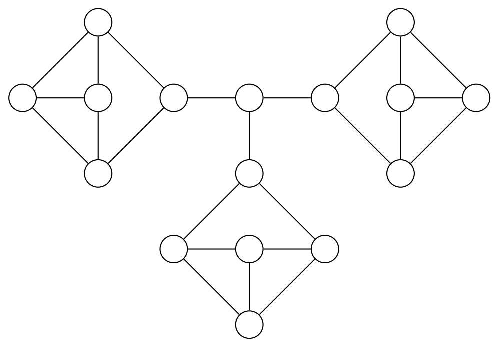
Clique
Eingabe: Ein ungerichter Graph $G = (V, E)$ und eine natürliche Zahl k.
Frage: Gibt es in G eine Clique der Größe k, das heißt, gibt es k Knoten in G, die alle paarweise durch eine Kante verbunden sind?

Aufgabe 5: **Approximation** (10 Punkte)

In dieser Aufgabe geht es um einen Algorithmus für Independent Set auf Graphen, in denen jeder Knoten maximal drei Nachbarn hat. Betrachten Sie dazu den folgenden Algorithmus:

Solange noch ein Knoten im Graph existiert: Nimm einen beliebigen Knoten v mit minimalem Knotengrad in die Lösungsmenge und lösche v und alle Nachbarn von v aus dem Graph.

(a) Zeichnen Sie eine mögliche Lösung des Algorithmus für den folgenden Graph ein (wie die beliebige Auswahl getroffen wird, ist Ihnen überlassen). Sie brauchen Ihre Antwort nicht zu begründen. (3 P)



(b) Zeigen Sie, dass der Algorithmus immer eine 3-Approximation liefert, wenn jeder Knoten im Eingabegraph höchstens drei Nachbarn hat. (7 P)

Erinnerungen:

Independent Set
Eingabe: Ein ungerichter Graph $G = (V, E)$.
Aufgabe: Fine eine größtmögliche Menge $K \subseteq V$ von Knoten, sodass zwischen keinen zwei Knoten in K eine Kante in E existiert.

Ein Algorithmus liefert eine 3-Approximation für Independet Set, wenn die vom Algorithmus gefundene Lösung immer mindestens $\text{opt}/3$ Knoten enthält, wobei opt die Größe einer optimalen Lösung ist.