Berlin, 3. März 2016

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

Klausur Grundlagen der Algorithmik

(Niedermeier/Froese/Chen/Fluschnik, Wintersemester 2015/16)

1	
2	
3	
4	
5	
Σ	

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

Allgemeine Hinweise:

- Es sind keinerlei Hilfsmittel erlaubt.
- Benutzen Sie keinen Bleistift, sondern einen Kugelschreiber oder Füller in der Farbe schwarz oder blau.
- Beschriften Sie jedes Blatt mit Vor- und Nachnamen sowie Matrikelnummer.
- **Falls in der Aufgabenstellung nicht explizit ausgeschlossen, sind alle Antworten zu begründen! Antworten ohne Begründung erhalten 0 Punkte.**

Viel Erfolg!

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

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

Betrachten Sie die folgende Problemstellung.

Eine Menge von n Personen $p_1, \ldots, p_n$ möchte am Semesterende eine Prüfung ablegen. Zur Auswahl stehen m mögliche Termine $t_1, \ldots, t_m$. Jede Person p_i hat nur zu einer bestimmten Menge $T_i \subseteq \{t_1, \ldots, t_m\}$ von möglichen Terminen Zeit für eine Prüfung. Außerdem muss jede Person die Prüfung an genau einem Termin ablegen. Allerdings können Prüfungen in Gruppen von maximal drei Personen gleichzeitig abgelegt werden. Ist es möglich, jeder Person einen ihrer möglichen Prüfungstermine zuzuordnen, sodass an keinem Termin mehr als drei Personen geprüft werden?

Modellieren Sie obiges Problem als ein Maximum Flow-Problem, sodass die Anzahl der Knoten und die Anzahl der Kanten im Flussnetzwerk höchstens polynomiell in n und m ist. Geben Sie hierfür die Knoten, Kanten und Kantenkapazitäten Ihres Flussnetzwerks an und begründen Sie, wie die Antwort bestimmt werden kann.

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

Aufgabe 2: **Lineares Programmieren** (5+5 Punkte)

Geben Sie für die beiden folgenden Probleme eine Formulierung als ganzzahliges lineares Programm an (ohne Begründung).

Definieren Sie dazu die verwendeten Variablen, alle Nebenbedingungen sowie die Zielfunktion und geben Sie an, ob diese minimiert oder maximiert wird.

(a) **Dominating Set**
Eingabe: Ein ungerichteter Graph G.
Aufgabe: Wähle möglichst wenige Knoten in G aus, sodass jeder andere Knoten mindestens einen dieser Knoten als Nachbarn hat.

(b) **Set Cover**
Eingabe: Eine Menge $\mathcal{C}$ von Teilmengen einer Grundmenge U.
Aufgabe: Wähle eine kleinstmögliche Teilmenge $\mathcal{C}' \subseteq \mathcal{C}$, sodass $\bigcup_{C \in \mathcal{C}'} C = U$.

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

Aufgabe 3: **Heuristiken** (3+3+5 Punkte)

Wir betrachten das folgende Problem.

> **WEIGHTED INDEPENDENT SET AUF BÄUMEN**
> **Eingabe:** Ein Baum $T = (V, E)$ mit Knotengewichten $w : V \to \mathbb{N}$.
> **Aufgabe:** Finde eine Knotenmenge $V' \subseteq V$ mit größtmöglichem Gesamtgewicht $\sum_{v \in V'} w(v)$, sodass keine zwei Knoten in V' durch eine Kante verbunden sind.

Gegeben sind folgende zwei Algorithmen zum Lösen des obigen Problems.

1. Wähle einen beliebigen Knoten als Wurzel. Nimm entweder alle Knoten aus jeder ungeraden Schicht oder alle Knoten aus jeder geraden Schicht in die Lösung V', je nachdem, welcher Fall das höhere Gesamtgewicht liefert.
2. Wähle einen beliebigen Knoten als Wurzel. Solange es noch mindestens zwei Knoten gibt, wähle ein Blatt mit maximaler Entfernung zur Wurzel und füge es samt seinen Geschwisterknoten zur Lösung V' hinzu. Lösche daraufhin das Blatt, seine Geschwisterknoten und seinen Elternknoten. Wenn ein einzelner Knoten übrig bleibt, nimm diesen in die Lösung V'.

(a) Zeigen Sie für beide Algorithmen, dass diese **nicht immer** eine optimale Lösung finden. Geben Sie dazu jeweils ein Beispiel mit einer optimalen Lösung und einer vom Algorithmus gefundenen suboptimalen Lösung an.

(b) Welcher der Algorithmen liefert für den Fall, dass alle Knoten $v \in V$ dasselbe Gewicht $w(v) = c > 0$ haben, immer eine optimale Lösung? Begründen Sie Ihre Antwort.

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

Aufgabe 4: **P vs NP-schwer** (3+3+3 Punkte)

Von den nachfolgenden Problemen sind mindestens drei in P und mindestens drei NP-schwer. Wählen Sie drei Probleme aus und argumentieren Sie, wie diese in Polynomzeit gelöst werden können.

(a) **Eingabe:** Ein ungerichteter Graph G.
Frage: Hat G ein Vertex Cover der Größe höchstens 10?

(b) **Eingabe:** Eine aussagenlogische Formel F in konjunktiver Normalform, bei der alle Literale nichtnegiert sind.
Frage: Ist F erfüllbar?

(c) **Eingabe:** Eine Menge $\mathcal{C}$ von zweielementigen Teilmengen einer Grundmenge U und eine natürliche Zahl k.
Frage: Kann man höchstens k Elemente aus U so auswählen, dass jede Teilmenge in $\mathcal{C}$ mindestens eins dieser Elemente enthält?

(d) **Eingabe:** Ein ungerichteter Graph G mit n Knoten.
Frage: Hat G eine Clique der Größe $n - 7$?

(e) **Eingabe:** Ein ungerichteter Graph G mit maximalem Knotengrad 3 und eine natürliche Zahl k.
Frage: Hat G ein Independent Set der Größe k?

(f) **Eingabe:** Ein Baum T, eine Menge $\{\{s_1, t_1\}, \ldots, \{s_r, t_r\}\}$ von Knotenpaaren aus T und eine Zahl k.
Frage: Ist es möglich, höchstens k Kanten aus T zu löschen, sodass keines der Knotenpaare mehr über einen Pfad verbunden ist?

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

Aufgabe 5: **Datenreduktion** (2+2+2+2+2 Punkte)

Das Max-2-SAT-Problem ist wie folgt definiert:

Max-2-SAT

Eingabe: Eine aussagenlogische Formel F in konjunktiver Normalform mit m Klauseln, die jeweils maximal zwei Literale enthalten und eine natürliche Zahl k.

Frage: Gibt es eine Variablenbelegung, durch die mindestens k Klauseln erfüllt werden?

Hinweis: Klauseln sind durch „oder" ($\vee$) verknüpfte Literale. Literale sind nichtnegierte oder negierte Variablen. Zum Beispiel ist $(x_1 \vee \overline{x_2})$ eine Klausel mit zwei Literalen.

Nachfolgend sind einige (vermeintlich korrekte) Datenreduktionsregeln für Max-2-SAT angegeben. Eine Datenreduktionsregel ist korrekt, wenn die resultierende Instanz genau dann eine „Ja"-Instanz ist, wenn auch die ursprüngliche Instanz eine „Ja"-Instanz ist.

Entscheiden Sie jeweils über die Korrektheit der vorgeschlagenen Regeln. Begründen Sie Ihre Antworten.

Wir nehmen an, dass in einer Klausel eine Variable höchstens einmal vorkommt.

(a) Gibt es eine Klausel bestehend aus nur einem Literal, so belege die zugehörige Variable so, dass die Klausel erfüllt wird, und verringere k um 1.

(b) Kommen „Monoklauseln" (x) und $(\overline{x})$ vor, so lösche beide Klauseln und verringere k um 1.

(c) Kommt eine Variable x ausschließlich nichtnegiert vor, so belege diese Variable mit wahr, verringere k um die Anzahl der Klauseln, in denen x vorkommt, und lösche diese Klauseln.

(d) Die Variablen x, y und z kommen ausschließlich in den Klauseln $(x \vee y)$, $(\overline{y} \vee z)$ und $(\overline{z} \vee \overline{x})$ vor. Dann lösche die drei Klauseln und verringere k um 3.

(e) Die Variable x kommt ausschließlich in den Klauseln $c_1 := (x \vee y)$ und $c_2 := (\overline{x})$ vor. Dann ersetze c_1 durch (y) und c_2 durch $(\overline{y})$ und lasse k unverändert.