2. Klausur zur Vorlesung
Algorithmentechnik
Wintersemester 2009/2010

Hier Aufkleber mit Name und Matrikelnummer anbringen

Vorname: ______________________

Nachname: ______________________

Matrikelnummer: ______________________

Beachten Sie:

- Bringen Sie den Aufkleber mit Ihrem Namen und Matrikelnummer auf diesem Deckblatt an und beschriften Sie jedes Aufgabenblatt mit Ihrem Namen und Matrikelnummer.
- Schreiben Sie die Lösungen auf die Aufgabenblätter und Rückseiten. Zusätzliches Papier erhalten Sie bei Bedarf von der Aufsicht.
- Zum Bestehen der Klausur sind **20** der möglichen **60** Punkte hinreichend.
- Es sind keine Hilfsmittel zugelassen.

Aufgabe	Mögliche Punkte					Erreichte Punkte				
	a	b	c	d	Σ	a	b	c	d	Σ
1	4	-	-	-	4		-	-	-	
2	2	3	1	-	6				-	
3	1	3	4	-	8				-	
4	2	1	4	-	7				-	
5	5	-	-	-	5		-	-	-	
6	2	2	3	2	9					
7	3	-	-	-	3		-	-	-	
8	1	3	4	-	8				-	
9	10x1				10					
Σ					60					

Problem 1: B-Zähler 4 Punkte

Ein B-Zähler ist ein Array $A[0 \dots k-1]$ mit $k \in \mathbb{N}$, $k > 0$ und $A[i] \in \{0, 1\}$ für $i = 0, \dots, k-1$. Zu Beginn sei $A[i] = 0$ für alle $i = 0, \dots, k-1$. Wir interpretieren den Inhalt von A als eine binär dargestellte Zahl z, d.h.

$$z := \sum_{i=0}^{k-1} A[i] \cdot 2^i.$$

Die einzige erlaubte Operation ist ERHÖHE. Dabei wird z um 1 erhöht. Das Array sei so implementiert, dass gilt:

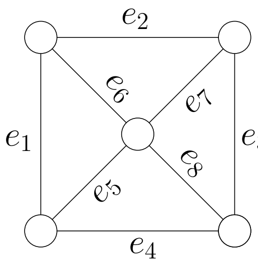
- Das **Setzen einer 1** an Stelle i des Arrays verursacht **Kosten von $i+1$**.
- Das Setzen beliebig vieler Stellen des Arrays auf 0 verursacht konstante Kosten von 1.

Zeigen Sie, dass die amortisierten Kosten für n ERHÖHE-Operationen in $O(n)$ liegen. Setzen Sie voraus, dass die Länge des Arrays für die Darstellung der Zahl $z + n$ ausreicht.

Problem 2: Kreisbasen $2 + 3 + 1 = 6$ Punkte

(a) Untenstehende Abbildung zeigt den Graphen $G := (V, E)$. Alle Kanten in G haben Gewicht 1.

Geben Sie eine **minimale Kreisbasis** B von G an. Stellen Sie die Basiskreise als Teilmengen von E dar. Zeigen Sie außerdem, dass die von Ihnen angegebene Kreismenge wirklich eine minimale Kreisbasis ist (zu zeigen sind die Minimalität *UND* die Basis-Eigenschaft).



(b) Widerlegen Sie folgende Aussage:
Es gibt eine **minimale Kreisbasis** zu G, die von mehreren verschiedenen aufspannenden Bäumen induziert wird.

(c) Erweitern Sie G durch das Einfügen von Kanten (und eventuell Knoten) so zu G', dass gilt:
Es gibt eine Fundamentalbasis zu G', die von mehreren verschiedenen aufspannenden Bäumen induziert wird.
Zeichnen Sie den erweiterten Graphen G' unter diesen Aufgabenteil.

Problem 3: Matroide $\quad 1 + 3 + 4 = 8$ Punkte

Gegeben sei ein Matroid $(M,\mathcal{U})$ mit dem zugehörigen Basissystem $\mathcal{B}$.
Weiterhin sei $w : M \to \mathbb{R}^+$ eine Gewichtsfunktion, die jedem Element eine *echt positive* Zahl zuordnet.

Das Problem, eine Basis $I^* \in \mathcal{B}$ zu finden, so dass $w(I^*)$ maximal ist unter allen Elementen aus $\mathcal{B}$, heißt *Maximierungsproblem*.

(a) Geben Sie die Greedy-Methode für das Maximierungsproblem an.

(b) Seien nun die Gewichte aller Elemente aus M paarweise verschieden.

Zeigen Sie: Die Optimallösung I^* für das Maximierungsproblem ist eindeutig.

(c) Gegeben sei ein ungerichteter, einfacher, vollständiger Graph $G = (V, E)$ mit $|V| \geq 3$.
Das heißt, für alle $u, v \in V$ ist $\{u, v\} \in E$ genau dann, wenn $u \neq v$.

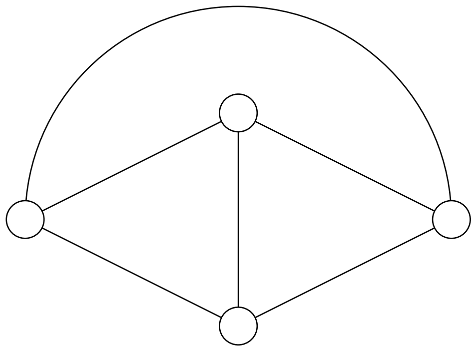
Wir definieren das Mengensystem $(E, \mathcal{U})$ mit

$$\mathcal{U} := \{I \subseteq E \mid I \text{ ist Teilmenge eines Hamiltonkreises in } G\}.$$

Ein Hamiltonkreis ist dabei ein einfacher Kreis in G, der jeden Knoten genau einmal enthält.

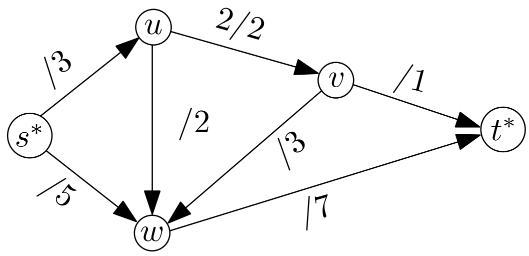
Zeigen Sie: $(E, \mathcal{U})$ ist ein Unabhängkeitssystem.

Zeigen Sie an folgendem Gegenbeispiel, dass $(E, \mathcal{U})$ im Allgemeinen kein Matroid ist.
Begründen Sie Ihre Antwort.



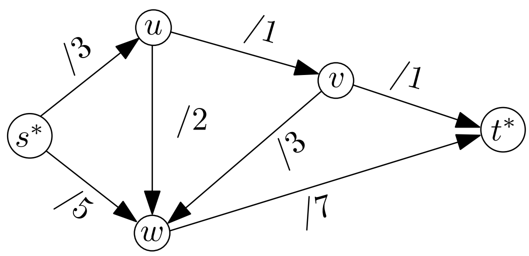
Problem 4: Flussnetzwerke 2 + 1 + 4 = 7 Punkte

(a) Gegeben sei das untenstehende Netzwerk $(D^*; s^*, t^*; c^*)$, die Kantenkapazitäten $c^*(e)$ sind an den Kanten notiert.



Zeichnen Sie in obiger Abbildung einen maximalen s^*-t^*-Fluss ein. Ergänzen Sie hierzu die Kantenbeschriftung im Format $f(e)/c^*(e)$. Die Kapazität der Kante (u, v) soll dabei **voll ausgenutzt** werden. Beweisen Sie die Maximalität Ihres Flusses, indem Sie einen minimalen Schnitt einzeichnen.

(b) Nun verringere sich die Kapazität der Kante (u, v) um 1. Verändern Sie den nun ungültigen Fluss aus (a), so dass wieder ein gültiger maximaler s^*-t^*-Fluss entsteht. Tragen Sie die neuen Flusswerte in untenstehende Abbildung ein. (Hinweis: Versuchen Sie den in u entstandenen Überschuss umzuleiten).

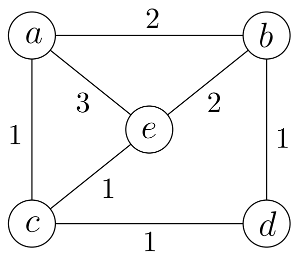


(c) Gegeben sei nun ein beliebiges Netzwerk $(D; s, t; c)$ mit ganzzahligen Kantenkapazitäten. In $(D; s, t; c)$ existiere eine Kante (s, v) mit ganzzahliger Kapazität $c(s, v) \leq 3$. Weiterhin sei ein maximaler, ganzzahliger s-t-Fluss f gegeben. Nun wird die Kante (s, v) gelöscht.

Beschreiben Sie die Arbeitsweise eines Algorithmus, der nach der Kantenlöschung in $O(|V| + |E|)$-Zeit einen neuen maximalen s-t-Fluss berechnet (kein Code, kein Pseudocode nötig). Begründen Sie die Korrektheit und die Laufzeit.

Problem 5: Stoer-Wagner-Schnitt-Algorithmus 5 Punkte

Wenden Sie auf den nebenstehend abgebildeten Graphen den Algorithmus von Stoer und Wagner an. Die Kantengewichte sind an den Kanten notiert. Geben Sie nach jeder Phase die Knoten s und t, den Schnitt der Phase und dessen Gewicht an. Zeichnen Sie den nach dem Verschmelzen resultierenden Graphen mit Kantengewichten. Der Startknoten in jeder Phase enthalte a. Geben Sie zum Schluss den minimalen Schnitt $S_{\min}$ an.



Phase 1:

$s =$ $t =$ Graph:

Schnitt der Phase $S_1 =$

$c(S_1, V \setminus S_1) =$

Phase 2:

$s =$ $t =$ Graph:

Schnitt der Phase $S_2 =$

$c(S_2, V \setminus S_2) =$

Phase 3:

$s =$ $t =$ Graph:

Schnitt der Phase $S_3 =$

$c(S_3, V \setminus S_3) =$

Phase 4:

$s =$ $t =$ Graph:

Schnitt der Phase $S_4 =$

$c(S_4, V \setminus S_4) =$

Minimaler Schnitt $S_{\min} =$

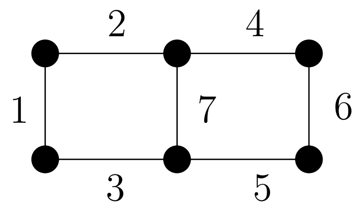
Problem 6: Maximum Spanning Tree $2+2+3+2=9$ Punkte

Gegeben sei ein zusammenhängender, ungerichteter Graph $G=(V,E,c)$ mit paarweise verschiedenen, positiven Kantengewichten $c: E \to \mathbb{R}^+$. Weiter bezeichne $\max(v) \in E$ die maximal gewichtete Kante unter allen zu $v \in V$ inzidenten Kanten. Es sei $S := \{\max(v) \mid v \in V\}$.

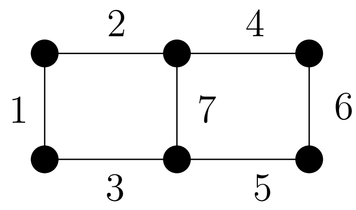
(a) Ein *Spannbaum* in G ist ein Baum $T := (V, E_T)$ mit $E_T \subseteq E$. Ein *maximaler* Spannbaum hat maximales Gewicht unter allen Spannbäumen in G. Zeichen Sie in untenstehenden Beispielgraphen G^* Folgendes ein:

- in (i) die Menge S.
- in (ii) einen maximalen Spannbaum.

Zeichnen Sie hierzu die Kanten jeweils fett ein. Die Kantengewichte sind an den Kanten notiert.



(i) Menge S.



(ii) maximaler Spannbaum.

(b) Sei $T_{\max} = (V, E_{\max})$ ein maximaler Spannbaum in G.
Zeigen Sie, dass $S \subseteq E_{\max}$ gilt.

(c) Zeigen Sie, dass $c(E_{\max}) \leq 2\,c(S)$ gilt.

(d) Für G bezeichne $\mathrm{OPT}(G)$ das Gewicht eines maximalen Spannbaums in G. Zeigen Sie, dass für untenstehenden Algorithmus $\mathcal{A}$ gilt:

$$\mathrm{OPT}(G) \leq 2 \cdot \mathcal{A}(G),$$

mit $\mathcal{A}(G) := c(E_T)$. (Hinweis: Man kann Teilaufgabe (c) nutzen.)

Algorithmus 1 : Approx-Max-Spannbaum

Eingabe : Graph $G = (V, E)$
Ausgabe : Baum $T = (V, E_T)$

$S \leftarrow \emptyset$
für alle $v \in V$ **tue**
  $S \leftarrow S \cup \{\max(v)\}$
$E_T \leftarrow S$
solange $|E_T| < n - 1$ **tue**
  Wähle $e \in E \setminus E_T$ beliebig, so dass e mit E_T keinen Kreis erzeugt
  $E_T \leftarrow E_T \cup \{e\}$
return $T = (V, E_T)$

Problem 7: Randomisierter Schnitt-Algorithmus 3 Punkte

Gegeben sei ein einfacher, ungerichteter Graph $G = (V, E)$.

Folgender randomisierter Algorithmus berechnet einen Schnitt $(S, V \setminus S)$ in G.

Algorithmus 2 : RandomCut

Eingabe : Einfacher, ungerichteter Graph $G = (V, E)$.
Ausgabe : Schnitt $(S, V \setminus S)$ mit $S \subseteq V$.

```
S ← ∅
für alle v ∈ V tue
    r ← rand({0,1}) ;                    // Wähle r zufällig gleichverteilt aus {0,1}
    Wenn r = 1
        S ← S ∪ {v}
return (S, V \ S)
```

Berechnen Sie die erwartete Anzahl an Kanten aus E, die den Schnitt $(S, V \setminus S)$ kreuzen.

Problem 8: Kernbildung für MAXSAT 1 + 3 + 4 = 8 Punkte

Das MAXSAT-Problem aus der Vorlesung ist wie folgt definiert.

Gegeben: Eine Menge von Klauseln C über der Variablenmenge V und ein Parameter k.
Frage: Gibt es eine Wahrheitsbelegung der Variablen aus V, so dass mindestens k Klauseln erfüllt werden?

(a) Betrachten Sie folgende Instanz:

$$V = \{x, y, z\}, \quad C = \{(x \vee \neg z), (\neg x \vee \neg y \vee \neg z), (\neg x \vee \neg y), (x \vee y \vee z)\}, \quad k = 3.$$

Geben Sie eine Wahrheitsbelegung $\beta : V \to \{\texttt{true}, \texttt{false}\}$ an, so dass mindestens 3 Klauseln aus C erfüllt werden.

$$\beta(x) = \qquad \beta(y) = \qquad \beta(z) =$$

(b) Sei k beliebig mit $1 \leq k \leq |C|$.

Sei $C_{\text{large}} \subseteq C$ die Teilmenge der Klauseln aus C, die mindestens k verschiedene Variablen enthalten.

Zeigen Sie: Ist $|C_{\text{large}}| \geq k$, dann gibt es eine Wahrheitsbelegung der Variablen in V, so dass mindestens k Klauseln aus C erfüllt werden.

(c) Sei $k \leq \lfloor |C|/2 \rfloor$.

Geben Sie einen Algorithmus mit polynomieller Laufzeit in der Anzahl $|V| + |C|$ der Variablen und Klauseln an (kein Pseudocode erforderlich), der eine Wahrheitsbelegung der Variablen findet, so dass mindestens k Klauseln aus C erfüllt sind.

Problem 9: $10 \times 1 = 10$ Punkte

Kreuzen Sie für folgende Aussagen an, ob diese wahr oder falsch sind. Für jede richtige Antwort gibt es einen Punkt, für jede falsche Antwort wird ein Punkt abgezogen. Es wird keine negative Gesamtpunktzahl für diese Aufgabe geben.

Sei (PP) ein lineares Programm und (DP) das zu (PP) duale Programm. Hat (PP) eine zulässige Lösung, dann hat auch immer (DP) eine zulässige Lösung. ☐ Wahr ☐ Falsch

Gegeben sei das CRCW-PRAM Modell, wobei mehrere Prozessoren genau dann gleichzeitig in eine Speicherstelle schreiben dürfen, wenn sie denselben Wert schreiben wollen. In diesem Modell gibt es einen Algorithmus, der das logische Oder von n booleschen Variablen mit n Prozessoren in Zeit $\mathcal{O}(1)$ berechnen kann. ☐ Wahr ☐ Falsch

Sei A die Array-Darstellung eines Heaps. Dann ist A entweder nicht-absteigend oder nicht-aufsteigend sortiert. ☐ Wahr ☐ Falsch

Seien B_1 und B_2 verschiedene minimale Spannbäume eines ungerichteten, gewichteten Graphen G und e eine Kante maximalen Gewichts von B_1. Dann gibt es eine Kante $e' \in G \setminus B_1$, so dass $B_2 = B_1 \setminus \{e\} \cup \{e'\}$ gilt. ☐ Wahr ☐ Falsch

Sei $(D; s, t; c)$ ein Flussnetzwerk. Bevor der Algorithmus von Goldberg und Tarjan ausgeführt wird, kann für das Label $\text{dist}(s)$ der Quelle s im Allgemeinen keine obere Schranke angegeben werden. ☐ Wahr ☐ Falsch

Sei P ein ILP und sei M das Lösungspolyeder zur Relaxierung von P. Falls M nicht leer ist, dann sind alle Ecken von M ganzzahlig. ☐ Wahr ☐ Falsch

Sei $G = (V, E)$ ein gewichteter, ungerichteter, einfacher Graph und e eine minimalgewichtete Kante aus E. Dann ist e in jedem MST enthalten. ☐ Wahr ☐ Falsch

Es gibt eine (unendliche) Familie $(G_i)_{i \in \mathbb{N}}$ von Graphen, so dass für jeden Graphen G_i die Anzahl der Kreisbasen in G_i exponentiell in der Anzahl der Knoten von G_i ist. ☐ Wahr ☐ Falsch

Sei Π ein Problem, k ein Parameter und n die Eingabelänge. Sei $\mathcal{A}$ ein Lösungsalgorithmus für Π, der eine erschöpfende Suche in einem binären Suchbaum der Tiefe 2^k durchführt. Weiterhin sei der Aufwand pro Baumknoten $p(n)$, wobei $p(n)$ ein Polynom ist, das ausschließlich von n abhängt. Dann ist Π fixed-parameter-tractable. ☐ Wahr ☐ Falsch

Jeder $\mathcal{RP}$-Algorithmus ist ein $\mathcal{BPP}$-Algorithmus. ☐ Wahr ☐ Falsch