1. Klausur zur Vorlesung
Algorithmentechnik
Wintersemester 2006/2007
1. März 2007

Hier Aufkleber mit Name und Matrikelnr. anbringen

Vorname: ______________________

Nachname: ______________________

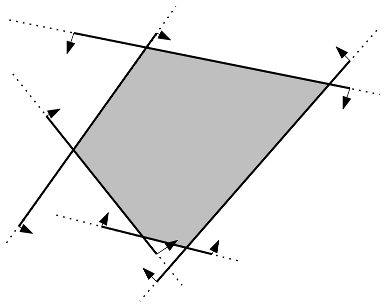
Matrikelnummer: ______________________

Beachten Sie:

- Bringen Sie den Aufkleber mit Ihrem Namen und Matrikelnummer auf diesem Deckblatt an und beschriften Sie jedes Aufgabenblatt mit Ihren Namen und Ihrer 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	1	2	3	4	5	6	7	8	9
a	1	4	2	2	1	2	2	2	–
b	3	2	2	3	1	1	3	3	–
c	2	–	4	–	4	3	1	2	–
$\sum$	6	6	8	5	6	6	6	7	10
$\sum$	60								
a									–
b									–
c		–		–					–
$\sum$									
$\sum$									

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



Gegeben seien n Halbebenen in der Ebene (siehe Bild rechts). Der folgende rekursive Algorithmus 1 berechnet den Schnitt dieser Halbebenen (graues Polyeder im Bild).
Hinweis: Die Routine SCHNEIDEKONVEXEREGIONEN(C_1, C_2) (Zeile 7) berechnet den Schnitt zweier Polyeder in einer Worst-case-Laufzeit von $\Theta(n^2)$.

Algorithmus 1 : HALBEBENENSCHNITT(H)

Eingabe : Menge H von n Halbebenen in der Ebene
Ausgabe : Das konvexe Polyeder $C := \bigcap_{h \in H} h$.

1 **Wenn** $|H| = 1$
2 $\quad C \leftarrow$ eindeutiges $h \in H$
3 **sonst**
4 $\quad$ Teile H in Mengen H_1 und H_2 mit Kardinalitäten $\lceil n/2 \rceil$ und $\lfloor n/2 \rfloor$
5 $\quad C_1 \leftarrow$ HALBEBENENSCHNITT(H_1)
6 $\quad C_2 \leftarrow$ HALBEBENENSCHNITT(H_2)
7 $\quad C \leftarrow$ SCHNEIDEKONVEXEREGIONEN(C_1, C_2)
8 Gib C aus

(a) Welchen Sinn erfüllen die Zeilen 1 und 2?

(b) Geben Sie die Rekursionsgleichung für die Laufzeit von Algorithmus 1 an. Begründen Sie diese knapp.

(c) Lösen Sie die Rekursionsgleichung aus Teilaufgabe (b) und geben Sie damit eine möglichst scharfe obere und untere Schranke für die Worst-case-Laufzeit an.

Problem 2: UNION-FIND $4 + 2 = 6$ Punkte

Das Problem ZUSAMMENHANGSKOMPONENTEN lautet wie folgt:
Gegeben: Ein ungerichteter, ungewichteter, einfacher Graph $G = (V, E)$.
Gesucht: Die Zusammenhangskomponenten von G.

(a) Schreiben Sie einen Algorithmus in Pseudocode zur Lösung des Problems ZUSAMMENHANGSKOMPONENTEN. Benutzen Sie dabei die aus der Vorlesung bekannte Datenstruktur UNION-FIND.

Algorithmus 2 : FINDE-ZUSAMMENHANG

Eingabe : Graph $G = (V, E)$ (ungerichtet, ungewichtet, einfach)
Ausgabe : UNION-FIND-Datenstruktur, in der Mengen Zusammenhang darstellen

(b) Geben Sie eine möglichst scharfe asymptotische obere Schranke für die Worst-case-Laufzeit Ihres Algorithmus an und begründen Sie diese.

Problem 3: Minimaler Spannbaum $2 + 2 + 4 = 8$ Punkte

(a) Schreiben Sie (umgangssprachlich) die Grüne Regel und die Rote Regel der Färbungsmethode von Tarjan auf (kein Pseudocode).

- Grüne Regel:

- Rote Regel:

[bitte wenden]

Gegeben sei der folgende Algorithmus 3.

Algorithmus 3 : SlowMST

Eingabe : Graph $G = (V, E)$ (ungewichtet, ungerichtet, zusammenhängend, einfach), injektive Kantengewichtsfunktion $w : E \to \mathbb{N}$

Ausgabe : Kantenmenge T

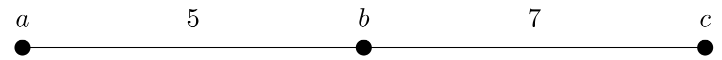
1 $T \leftarrow E$
2 $F \leftarrow \emptyset$
3 **solange** $|T| > n - 1$ **tue**
4     $C \leftarrow$ Kreis in $H = (V, T)$
5     $e \leftarrow$ beliebige Kante auf C
6     $T \leftarrow T \setminus \{e\}$
7     $F \leftarrow F \cup \{e\}$
8 **solange** $F \neq \emptyset$ **tue**
9     $e \leftarrow$ beliebige Kante aus F
10     $F \leftarrow F \setminus \{e\}$
11     $T \leftarrow T \cup \{e\}$
12     $C \leftarrow$ Kreis in T
13   **Wenn** *e Kante auf C mit maximalem Gewicht*
14         $T \leftarrow T \setminus \{e\}$
15   **sonst**
16         $e' \leftarrow$ beliebige Kante auf C mit $w(e') > w(e)$
17         $T \leftarrow T \setminus \{e'\}$
18         $F \leftarrow F \cup \{e'\}$
19 Gib T aus

(b) Was genau induziert T unmittelbar vor der ersten Ausführung von Zeile 8?

(c) Begründen Sie, dass Algorithmus 3 einen MST ausgibt (kein vollständiger Beweis erforderlich).

Problem 4: Minimaler Schnitt $2 + 3 = 5$ Punkte

(a) Führen Sie den Algorithmus von Stoer und Wagner auf untenstehendem Graphen durch. Startknoten sei immer Knoten a. Geben Sie die geforderten Informationen an und zeichnen Sie den Graphen nach jeder Phase (also nach dem Verschmelzen).



Phase 1

$s =$ $\quad\quad, t =$

Schnitt der Phase: $(\{\quad\},\{\quad\})$

Wert des Schnittes aus Phase 1:

Graph nach Phase 1:

Phase 2

$s =$ $\quad, t =$

Schnitt der Phase: $(\{\quad\},\{\quad\})$

Wert des Schnittes aus Phase 2:

Graph nach Phase 2:

Ergebnis

Resultierender minimaler Schnitt $(\{\quad\},\{\quad\})$,

mit Wert:

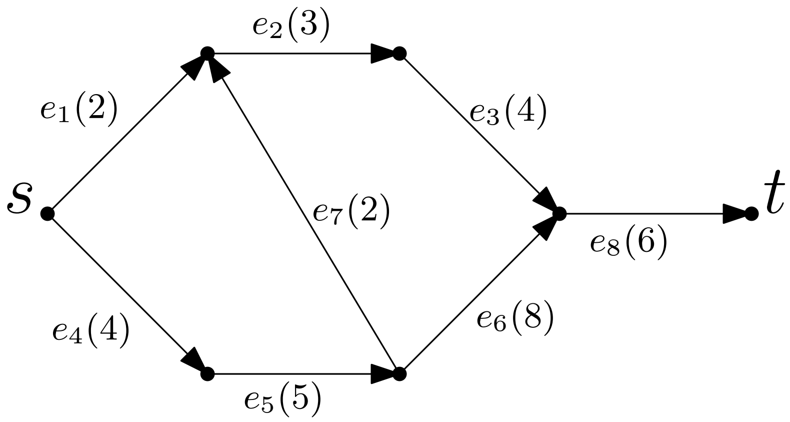
[bitte wenden]

(b) Zeigen Sie: In einem (zusammenhängenden, einfachen, ungerichteten) Graphen $G = (V, E)$ mit Kantengewichtsfunktion $w : E \to \mathbb{Z}$ (negative Kantengewichte zugelassen) arbeitet der Algorithmus von Stoer und Wagner nicht notwendigerweise korrekt.
Hinweis: Schauen Sie sich Teilaufgabe (a) scharf an.

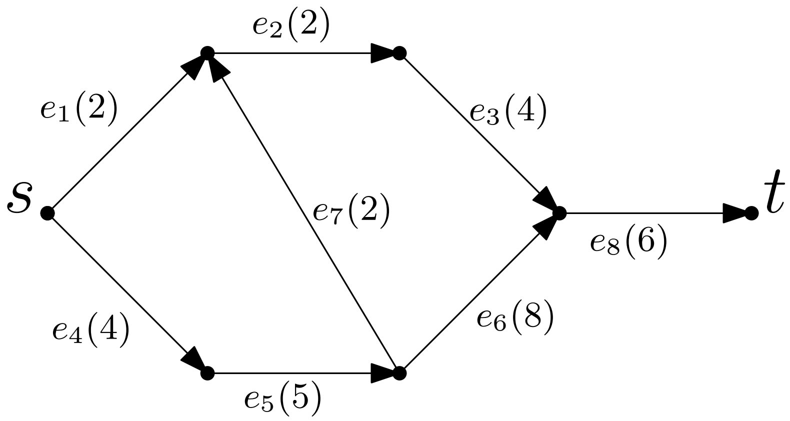
Problem 5: Maximaler Fluss $1 + 1 + 4 = 6$ Punkte

Gegeben sei das folgende Netzwerk D mit ganzzahligen Kantenkapazitäten (in Klammern) wie angegeben.

(a) Zeichnen Sie den maximalen Fluss von s nach t ein, indem Sie den Wert neben die entsprechende Kante in der Abbildung schreiben. Dabei soll die Kapazität der Kante e_2 voll ausgenutzt werden.



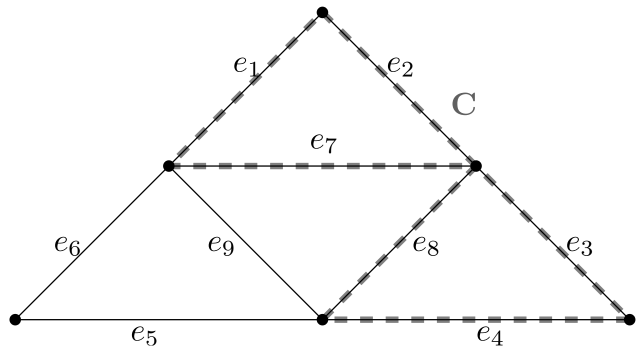
(b) Die Kapazität der Kante e_2 verringere sich nun um 1 auf die neue Kapazität 2. Korrigieren Sie Ihren Fluss aus Teilaufgabe (a), so dass wieder ein gültiger maximaler Fluss entsteht, und zeichnen Sie diesen im untenstehenden Netzwerk ein.



(c) Gegeben sei ein maximaler Fluss in einem Netzwerk mit ganzzahligen Kantenkapazitäten. Beschreiben Sie ein Verfahren, das diesen Fluss in einen gültigen maximalen Fluss überführt, nachdem sich die Kapazität einer Kante $e = (v, w)$ um 1 verringert hat (kein Code, kein Pseudocode). Nehmen Sie an, dass die Kapazität vor der Verringerung echt positiv war.

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

Die folgende Abbildung zeigt den Graphen G_{Tri}. Alle Kantengewichten seien 1. Gestrichelt grau ist zudem ein Kreis C in G_{Tri} eingezeichnet.



(a) Geben Sie eine minimale Kreisbasis $\mathcal{B}$ von G_{Tri} an (mit Scharfblick oder Methode). Geben Sie das Gewicht von $\mathcal{B}$ an.

(b) Bilden Sie den Kreis C (gestrichelt, grau) durch eine Linearkombination ihrer Basiselemente aus $\mathcal{B}$.

(c) Begründen Sie, dass in G_{Tri} keine minimale Kreisbasis existiert, die eine Fundamentalbasis ist.

Problem 7: Randomisierte Algorithmen $2 + 3 + 1 = 6$ Punkte

Sei $n \in \mathbb{N}$. Gegeben seien drei Polynome $P_1(x), P_2(x), P_3(x)$ aus $\mathbb{R}[x]$, mit $\deg P_1 \leq n$, $\deg P_2 \leq n$ und $\deg P_3 \leq 2n$. Ferner sei $k \in \mathbb{N}$ gegeben. Zu prüfen ist, ob gilt: $P_1(x) \cdot P_2(x) = P_3(x)$.

Algorithmus 4 : PolynomTest$(P_1(x), P_2(x), P_3(x), k)$

Eingabe : $k \in \mathbb{N}$ und drei Polynome aus $\mathbb{R}[x]$: $P_1(x), P_2(x), P_3(x)$, mit $\deg P_1 \leq n$, $\deg P_2 \leq n$ und $\deg P_3 \leq 2n$

Ausgabe : Aussage, ob gilt: $P_1(x) \cdot P_2(x) = P_3(x)$

$S \leftarrow$ beliebige endliche Teilmenge aus $\mathbb{R}$ mit $|S| \geq 2n + 1$
Für $i \leftarrow 1$ *bis* k
    $r \leftarrow$ zufälliges Element aus S
    Wenn $P_1(r) \cdot P_2(r) \neq P_3(r)$
        Gib Nein aus
Gib Ja aus

(a) Zeigen Sie: Wenn $P_1(x) \cdot P_2(x) = P_3(x)$ gilt, dann gibt Algorithmus 4 das korrekte Ergebnis aus.

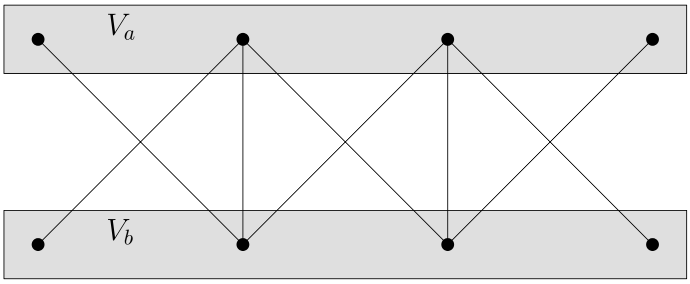
(b) Zeigen Sie: Die Wahrscheinlichkeit, dass Algorithmus 4 fälschlicherweise Ja ausgibt, ist kleiner gleich $(2n/|S|)^k$.

(c) Von welchem Typ randomisierter Algorithmen ist Algorithmus 4?
Begründen Sie Ihre Aussage kurz.

Problem 8: ILP für MAXIMALES BIPARTITES MATCHING $2 + 3 + 2 = 7$ Punkte

Das Problem MAXIMALES BIPARTITES MATCHING lautet wie folgt:
Gegeben: Ein ungerichteter, ungewichteter, bipartiter Graph $G = (V := V_a \dot{\cup} V_b, E)$ mit der Partition V_a, V_b.
Hinweis: In einem bipartiten Graphen ist $E \subseteq \{\{v, w\} \mid v \in V_a, w \in V_b\}$.
Gesucht: $M \subseteq E$ mit $|M|$ maximal so, dass jeder Knoten $v \in V$ zu maximal einer Kante aus M inzident ist (M ist dann ein *maximales Matching*).

(a) Zeichnen Sie in dem untenstehenden Graphen ein maximales Matching ein.



(b) Modellieren Sie das Problem als ILP. Erläutern Sie knapp Zielfunktion, Variablen und Nebenbedingungen. Was besagt der Wert der Zielfunktion einer Lösung?

(c) Ist das Problem MAXIMALES BIPARTITES MATCHING in der Komplexitätsklasse $\mathcal{P}$ enthalten? Begründen Sie Ihre Antwort.

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

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

Aussage	Wahr	Falsch
Zur Bestimmung eines MST ist auf Graphen mit $\|E\| \in O(\|V\|)$ der Algorithmus von Kruskal dem von Prim vorzuziehen (bzgl. Laufzeit), sofern die Kanten nach Gewicht sortiert vorliegen.	☐	☐
Es gilt: $o(n \cdot (\log n)^2) \cap O(n \cdot \log(n^2)) \cap \Omega(n \cdot \log\log n) \neq \emptyset$.	☐	☐
Auf einer CREW-PRAM kann das Produkt von n Zahlen in einer asymptotischen Laufzeit von $O(\log n)$ berechnet werden.	☐	☐
In einem ungerichteten, einfachen, zusammenhängenden Graphen ist der Kreisraum orthogonal zum Schnittraum.	☐	☐
Die Menge aller bipartiten Teilgraphen (induziert durch Kantenmengen) eines Graphen bilden ein Matroid über dem Kantenraum.	☐	☐
Wenn bei einer amortisierten Analyse für die Potentialfunktion $\mathbb{C}$ und zwei aufeinanderfolgende Zustände stets gilt: $\mathbb{C}(D_{i+1}) \geq \mathbb{C}(D_i)$, dann sind die amortisierten Kosten eine obere Schranke für die Gesamtkosten.	☐	☐
Das Löschen des maximalen Elements aus einem MAX-HEAP mit n Elementen (mit Wiederherstellung der HEAP-Eigenschaft), hat eine Worst-case-Laufzeit von $\Theta(n)$ (Implementierung nach Vorlesung).	☐	☐
Bei der Ausführung des Algorithmus von GOLDBERG-TARJAN werden höchstens $O(\|V\| \cdot \|E\|)$ PUSH-Operationen durchgeführt (Standardimplementierung nach Vorlesung).	☐	☐
Die asymptotische Laufzeit eines FPAS ist polynomiell in der Eingabegröße und in $1/\epsilon$ (ϵ wie in Vorlesung).	☐	☐
Jeder parametrisierte Algorithmus für k-VERTEX-COVER hat eine Laufzeit, die exponentiell in $\|V\|$ ist.	☐	☐