2. Klausur zur Vorlesung
Algorithmentechnik
Wintersemester 2008/2009

Lösung!

Beachten Sie:

- Bringen Sie den Aufkleber mit Ihrem Namen und Matrikelnummer auf diesem Deckblatt an und beschriften Sie jedes Aufgabenblatt mit Ihren 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	e	Σ	a	b	c	d	e	Σ
1	4	-	-	-	-	4		-	-	-	-	
2	2	3	1	-	-	6				-	-	
3	2	1	4	-	-	7				-	-	
4	1	4	-	-	-	5			-	-	-	
5	2	3	2	-	-	7				-	-	
6	2	3	2	-	-	7				-	-	
7	2	4	-	-	-	6			-	-	-	
8	2	1	3	-	-	6				-	-	
9	12x1					12						
Σ						60						

Problem 1: Dynamische Arrays — 4 Punkte

Gegeben sei eine dynamische Datenstruktur A, die initial Kapazität $cap(A) = 2$ und Länge $len(A) = 0$ habe. Auf A sei eine Operation INSERT definiert, die ein Element in A einfügt, und folgende Effekte hat:

- **Fall 1 (len(A) < cap(A))**: $len(A) := len(A) + 1$ mit Kosten 1
- **Fall 2 (len(A) = cap(A))**: $len(A) := len(A) + 1$, $cap(A) := 2 \cdot cap(A)$ mit Kosten $cap(A)$

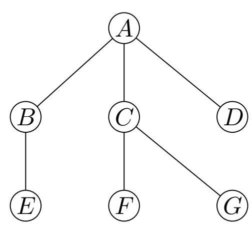
Zeigen oder widerlegen Sie: Die Kosten für n INSERT-Operationen liegen in $O(n)$.

Lösung: Die realen Kosten für die n-te INSERT-Operation bezeichnen wir mit $c(n)$. Wir benutzen die Buchungsmethode und setzen die amortisieren Kosten $\tilde{c}(n)$ pro INSERT-Operation auf 3. Somit betragen die amortisieren Kosten für n-Operationen $3n$, liegen also in $O(n)$. Es sind die amortisieren Kosten eine obere Schranke für die tatsächlichen: Für $n = 1, 2, 3$ kann man das leicht nachprüfen. Es gelte die Behauptung für alle $n = 1, \dots, 2^k + 1$. Dann gilt sie auch für alle $n = 2^k + 2, \dots 2^{k+1}$. denn dort sind die amortisieren Kosten 3 und die realen Kosten 1. Außerdem gilt es auch für $n = 2^{k+1}$, denn $\sum_{n=2^k+2}^{2^{k+1}+1} c(n) = (2^k - 2) \cdot 1 + 2^{k+1} = 3 \cdot 2^k - 2 \leq 3 \cdot 2^k = \sum_{n=2^k+2}^{2^{k+1}+1} \tilde{c}(n)$. Mit vollständiger Induktion folgt die Behauptung.

Problem 2: Union-Find — 2 + 3 + 1 = 6 Punkte

Im folgenden sei Union-Find jeweils mit Weighted Union und Pfadkompression implementiert, wie in der Vorlesung vorgestellt.

(a) Betrachten Sie den in der folgenden Darstellung abgebildeten Zustand einer Union-Find-Datenstruktur. Geben Sie eine Folge von Operationen MAKESET, UNION und FIND an, die den dargestellten Zustand der Datenstruktur erzeugen *kann*.



Lösung: Eine mögliche Folge von UNION-Operationen ist $(B, E), (A, D), (A, B), (C, F), (C, G), (A, C)$. Natürlich muss vorher für alle Elemente MAKESET aufgerufen worden sein.

(b) Gegeben sei ein Graph $G = (V, E)$ und eine Färbung der Knoten $c: V \to \mathcal{C}$, wobei $\mathcal{C}$ eine Menge von Farben sei.

- Eine Menge M von Knoten heißt *monochrom zusammenhängend*, wenn je zwei Knoten in M durch einen Weg verbunden sind, auf dem alle Knoten gleich gefärbt sind.
- Eine monochrom zusammenhängende Menge M heißt *maximal*, wenn es keine monochrom zusammenhängende Menge M' mit $M \subsetneq M'$ gibt.

Formulieren Sie mithilfe der Union-Find-Datenstruktur aus der Vorlesung einen Algorithmus, der die maximalen monochrom zusammenhängenden Mengen eines gegebenen Graphen $G = (V, E)$ berechnet.

Algorithmus 1 : Maximale monochrom zusammenhängende Mengen

Eingabe : Graph $G = (V, E)$, Knotenfärbung $c: V \to \mathcal{C}$
Ausgabe : Union-Find-Datenstruktur, so dass $\textsc{Find}(u) = \textsc{Find}(v)$ genau dann wahr ist, wenn u und v in der selben maximalen monochrom zusammenhängenden Menge liegen

Lösung:

Algorithmus 2 : Maximale monochrom zusammenhängende Mengen

Eingabe : Graph $G = (V, E)$, Knotenfärbung $c: V \to \mathcal{C}$
Ausgabe : Union-Find-Datenstruktur der maximalen monochrom zusammenhängenden Mengen

```
Für v ∈ V
    MakeSet(v)
Für {u, v} ∈ E
    Wenn Find(u) ≠ Find(v) und c(u) = c(v)
        Union(u, v)
```

(c) Geben Sie eine möglichst kleine obere Schranke für die asymptotische Laufzeit Ihres Algorithmus an. Begründen Sie Ihre Antwort.

Lösung: Die Laufzeit des Algorithmus ist in $\mathcal{O}(|V| + |E|\alpha(|V|, |E|))$, da nach $|V|$ MAKESET Operationen, die in konstanter Zeit ausgeführt werden können, lediglich noch $\mathcal{O}(|E|)$ Union- und Find-Operationen ausgeführt werden.

Problem 3: Matroide — 2 + 1 + 4 = 7 Punkte

(a) Gegeben sei die Grundmenge $M = \{a_1, a_2, a_3, a_4\}$ und das Mengensystem $U = \{\{a_1, a_2, a_3\}, \{a_3\}\}$. Geben Sie die Menge U_1 mit minimaler Kardinalität an, so dass $U \cup U_1$ ein Unabhängigkeitssystem ist. Geben Sie die Menge U_2 mit minimaler Kardinalität an, so dass $U \cup U_2$ ein Matroid ist.

Lösung:

$$U_1 = \{\emptyset, \{a_1\}, \{a_2\}, \{a_1, a_2\}, \{a_1, a_3\}, \{a_2, a_3\}\}$$

$$U_2 = U_1$$

(b) Beweisen Sie: In einem Matroid $\mathcal{M}$ haben alle maximalen unabhängigen Mengen die gleiche Kardinalität.

Lösung: Angenommen es gibt zwei maximale unabhängige Mengen U_1 und U_2 mit unterschiedlicher Kardinalität. Es sei ohne Beschränkung der Allgemeinheit $|U_2| < |U_1|$. Dann gilt die Austauscheigenschaft und es gibt ein Element $u \in U_1 \setminus U_2$ so dass $U_2 \cup \{u_1\}$ unabhängig ist. Dies ist ein Widerspruch zur Maximalität von U_2.

(c) Gegeben sei eine Menge M und zwei Elemente $e_1, e_2 \in M$. Es sei $\mathcal{U}$ das System der Teilmengen von M, die nicht gleichzeitig e_1 und e_2 enthalten (also das System $\mathcal{U}$ der Mengen K für die gilt $|K \cap \{e_1, e_2\}| \leq 1$). Zeigen Sie: $(M, \mathcal{U})$ ist ein Matroid.

Hinweis: Betrachten Sie für die Austauscheigenschaft Mengen $A, B \in \mathcal{U}$ mit $|A| < |B|$ und machen Sie eine Fallunterscheidung nach $|A \cap \{e_1, e_2\}|$.

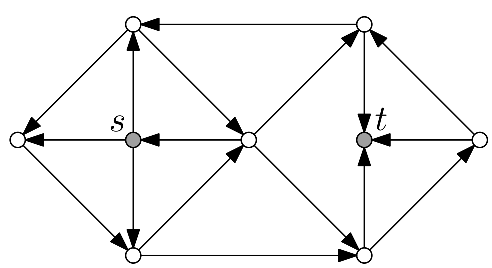
Lösung: Es gilt

- $\emptyset \in \mathcal{F}$
- Es ist $U_1 \in \mathcal{F}$ und $U_2 \subseteq U_1$ äquivalent mit $|U_1 \cap \{e_1, e_2\}| \leq 1$ und $U_2 \subseteq U_1$. Daraus folgt $|U_1 \cap \{e_1, e_2\}|$ und damit $U_2 \in \mathcal{F}$.
- Es seien $U_1, U_2 \in \mathcal{F}$ und $|U_1| = |U_2| + 1$. Wir unterscheiden folgende Fälle:
  (a) $e_1, e_2 \notin U_2$: Dann gilt für jedes $u \in U_1 \setminus U_2$ dass $U_2 \cup \{x\} \in \mathcal{F}$.
  (b) Falls $e_2 \notin U_1$ so ist $U_1 \setminus U_2$ nichtleer und für jedes x in $U_1 \setminus U_2$ gilt $U_2 \cup \{x\} \in \mathcal{F}$. Falls $e_2 \in U_1$ setzen wir $\overline{U} = (U_1 \setminus \{e_2\}) \setminus (U_2 \setminus \{e_1\})$. Es ist $\overline{U}$ nichtleer und für jedes $x \in \overline{U}$ gilt $U_1 \cup \{x\} \in \mathcal{F}$.

Problem 4: Flüsse und Zusammenhang $1 + 4 = 5$ Punkte

Sei $G = (V, E)$ ein gerichteter Graph mit ausgezeichneten Knoten s und t ($s \neq t$). Ein s-t-Weg in G ist ein gerichteter Weg von s nach t ohne Kreise.
Zwei s-t-Wege W_1 und W_2 heißen *kantendisjunkt*, wenn sie keine gemeinsamen Kanten haben. Die maximale Anzahl paarweise kantendisjunkter s-t-Wege in G sei mit $\kappa_G(s, t)$ bezeichnet.

(a) Zeichnen Sie eine maximale Anzahl paarweise kantendisjunkter s-t-Wege in den folgenden Graphen ein.



(b) Sei $f : E \to \mathbb{N}$ ein ganzzahliger maximaler s-t-Fluss in dem zu G gehörigen Flussnetzwerk $\mathcal{N}$ mit Kantenkapazität $c(u, v) = 1$ für alle $(u, v) \in E$. Sei $w(f)$ der Wert des Flusses.

Zeigen Sie, dass es mindestens $w(f)$ paarweise kantendisjunkte s-t-Wege in G gibt, d.h. dass $\kappa_G(s,t) \geq w(f)$ gilt.

Lösung: Offenbar ist $f(u,v) \in \{0,1\}$ für alle $(u,v) \in E$. Sei o.E. $w(f) > 0$ und sei W ein einfacher s-t-Weg, der nur Kanten (u,v) mit $f(u,v) = 1$ benutzt (für $w(f) > 0$ muss ein solcher Weg existieren). Sei $E(W)$ die Menge der Kanten auf W.

Betrachte die Abbildung $f' : E \to \mathbb{N}$, die definiert ist durch

$$f'(u,v) = \begin{cases} f(u,v), & \text{falls } (u,v) \notin E(W) \\ 0, & \text{sonst .} \end{cases}$$

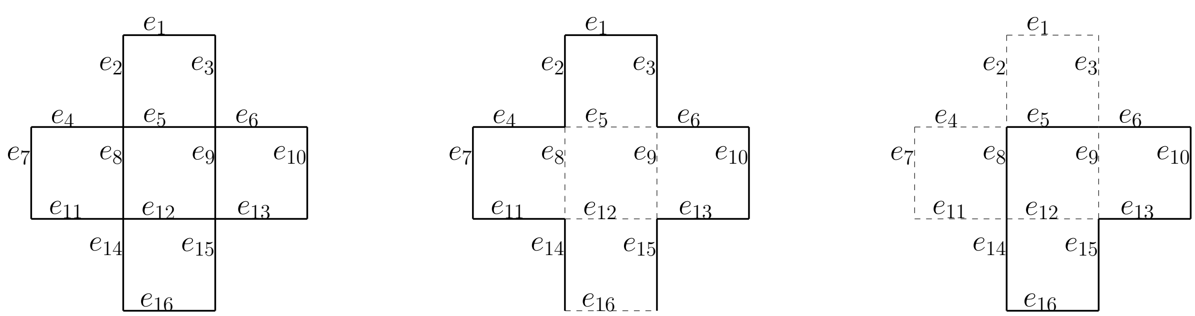
Wegen $f' \leq f$ erfüllt f' die Kapazitätsbedingung trivialerweise. Außerdem erfüllt f' die Flusserhaltungsbedingung, da f' aus f durch gleichmäßiges Absenken des Flusses von je einer eingehenden und einer ausgehenden Kante an den Knoten in $V(W) \setminus \{s,t\}$ hervorgeht. Also ist f' wieder ein Fluss.

Der Wert $w(f')$ ist genau $w(f) - 1$, da in s (bzw. in t) genau eine ausgehende (bzw. eingehende) Flusseinheit entfernt wird.

Dieser Vorgang kann $w(f)$ mal iteriert werden und liefert in jedem Schritt einen s-t-Weg, der keine Kanten mit bereits gefundenen s-t-Wegen gemeinsam hat. Damit gilt die Aussage.

Problem 5: Kreisbasen — 2 + 3 + 2 = 7 Punkte

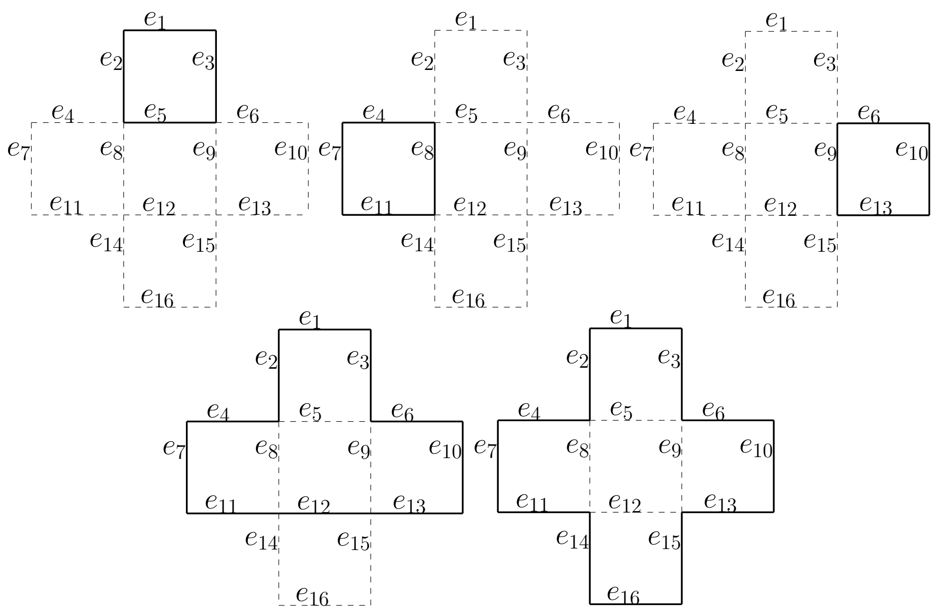
Gegeben sei der Graph $G = (V,E)$. Alle Kanten haben einheitliches Gewicht. Gegeben seien zusätzlich der Baum $T = \{e_1, e_2, e_3, e_4, e_6, e_7, e_{10}, e_{11}, e_{13}, e_{14}, e_{15}\}$ und der Kreis $C = \{e_5, e_6, e_8, e_{10}, e_{13}, e_{14}, e_{15}, e_{16}\}$ in G (wie in der folgenden Abbildung dargestellt).



Graph G — Baum T — Kreis C

(a) Geben Sie die Fundamentalkreisbasis B von G bezüglich T an und stellen Sie den den Kreis C als Linearkombination von Elementen in B dar.

Lösung:



$C = C_5 \oplus C_8 \oplus C_{16}$

(b) Alle Kanten in G haben einheitliches Gewicht. Geben Sie eine minimale Kreisbasis B von G an (genaues Hinsehen genügt). Begründen Sie kurz warum B eine minimale Kreisbasis ist.

Lösung: $B = \{B_1 = \{e_1, e_2, e_3, e_5\}, B_2 = \{e_4, e_7, e_8, e_{11}\}, B_3 = \{e_5, e_8, e_9, e_{12}\}, B_4 = \{e_6, e_9, e_{10}, e_{13}\}, B_5 = \{e_{12}, e_{14}, e_{15}, e_{16}\}\}$ ist eine MCB.

Die Dimension 5 ergibt sich aus Teilaufgabe a) oder der Formel $m - n + 1$. Die Kreise sind offensichtlich unabhängig, da sich kein Kreis als Linearkombination der anderen darstellen lässt. Jeder Kreis in G hat Länge mindestens 4. Also ist eine Kreisbasis mit Gewicht 20 minimal.

(c) Zeigen Sie, dass die symmetrische Differenz $C_1 \oplus C_2$ zweier allgemeiner Kreise C_1 und C_2 wieder ein allgemeiner Kreis ist.

Lösung:

$$\begin{aligned}
\deg_{C_1 \oplus C_2}(v) &= |\{\{u,v\} \in E \mid \{u,v\} \in C_1 \setminus (C_1 \cap C_2)\}| + |\{\{u,v\} \in E \mid \{u,v\} \in C_2 \setminus (C_1 \cap C_2)\}| \\
&= |\{\{u,v\} \in C_1\}| + |\{\{u,v\} \in C_2\}| - 2|\{\{u,v\} \in C_1 \cap C_2\}| \\
&= \deg_{C_1}(v) + \deg_{C_2}(v) - 2\deg_{C_1 \cap C_2}(v) \equiv 0 \mod (2)
\end{aligned}$$

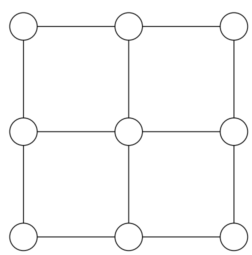
Problem 6: ILP — 2 + 3 + 2 = 7 Punkte

Gegeben sei ein Graph $G = (V, E)$. Für einen Knoten $u \in V$ sei die *Nachbarschaft einschließlich u* definiert als $N(u) := \{u\} \cup \{w \in V \mid \{u,w\} \in E\}$. Eine *dominierende Menge* $X \subseteq V$ ist eine Menge von Knoten mit folgender Eigenschaft:

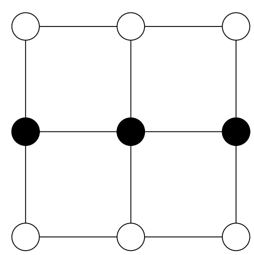
- Für alle Knoten $u \in V$ gibt es mindestens einen Knoten $w \in N(u)$ mit $w \in X$.

Das Problem DOMINIERENDEMENGE besteht darin eine dominierende Menge minimaler Kardinalität zu finden.

(a) Zeichnen Sie eine dominierende Menge minimaler Kardinalität in den folgenden Graphen ein.
Hinweis: Die optimale Lösung besteht aus weniger als 4 Knoten!



Lösung:



(b) Formulieren Sie das Problem DOMINIERENDEMENGE als ganzzahliges lineares Programm. Erklären Sie die Bedeutung der Variablen und Nebenbedingungen.

Lösung:

$$\min \sum_{u \in V} X_u \tag{1}$$

$$X_u \in \{0,1\} \qquad u \in V \tag{2}$$

$$\sum_{w \in N(u)} X_w \geq 1 \qquad u \in V \tag{3}$$

Es gibt je eine Variable X_u für alle $u \in V$. Die Variable X_u sei 1 genau dann, wenn u in der dominierenden Menge enthalten ist (0 sonst). Für jeden Knoten $u \in V$ gibt es eine Nebenbedingung, die sicherstellt, dass es einen Knoten w in der Nachbarschaft von u gibt, welches in X enthalten ist. Die Kardinalität der Menge $X := \{w \in V \mid X_w = 1\}$ wird minimiert, d.h. es wird eine minimale dominierende Menge gesucht.

(c) Gegeben sei ein Graph $G = (V, E)$. Betrachten Sie das folgende ganzzahlige lineare Programm für G:

$$\max \sum_{\{u,v\} \in E} X_{u,v} \tag{4}$$

$$X_{u,v} \in \{0, 1\} \qquad \text{für alle } \{u, v\} \in E \tag{5}$$

$$X_{u,v} + X_{v,w} \leq 1 \qquad \text{für alle } \{u, v\}, \{v, w\} \in E \text{ mit } u \neq w \tag{6}$$

Geben Sie jeweils eine sinnvolle Interpretation der Variablen und Nebenbedingungen an und interpretieren Sie die Zielfunktion des Programms entsprechend.

Lösung: Für jede Kante $\{u, v\} \in E$ gibt es eine Variable $X_{\{u,v\}}$, die genau dann 1 ist, wenn $\{u, v\}$ in der gesuchten Menge von Kanten enthalten ist (0 sonst). Für jeden Knoten v und je zwei adjazente Knoten $u \neq w$ gibt es eine Nebenbedingung, die sicherstellt, dass jeweils höchstens eine zu v inzidente Kante ausgewählt wird. Die Kardinalität der gesuchten Menge soll maximiert werden, d.h. es wird ein maximales Matching gesucht.

Problem 7: 3-Hitting-Set $\hfill$ 2 + 4 = 6 Punkte

Gegeben sei eine Grundmenge $S = \{1, \dots, n\}$ und ein Mengensystem $\mathcal{C} \subseteq 2^S$. Jede Menge $A \in \mathcal{C}$ enthalte maximal 3 Elemente, d.h. $|A| \leq 3$.
Ein Hitting Set für $\mathcal{C}$ ist eine Teilmenge $X \subseteq S$, so dass für jedes $A \in \mathcal{C}$ gilt $A \cap X \neq \emptyset$. Gesucht ist ein Hitting Set minimaler Kardinalität.

Betrachten Sie den folgenden Algorithmus:

Algorithmus 3 : 3-Hitting-Set-Approximation

Eingabe : $S = \{1, \dots, n\}$, $\mathcal{C} \subseteq 2^S$ mit $|A| \leq 3$ für alle $A \in \mathcal{C}$
Ausgabe : Hitting Set X für $\mathcal{C}$

1 $X \leftarrow \emptyset$
2 **solange** $\mathcal{C} \neq \emptyset$ **tue**
3 $\quad A \leftarrow$ wähle eine Menge $A \in \mathcal{C}$
4 $\quad X \leftarrow X \cup A$
5 $\quad \mathcal{C} \leftarrow \mathcal{C} \setminus \{A \in \mathcal{C} \mid A \cap X \neq \emptyset\}$

(a) Geben Sie die Menge X an, die Algorithmus 3 bei Eingabe der folgenden Instanz berechnet:

$$S = \{1, 2, 3, 4, 5\}, \quad \mathcal{C} = \{c_1 = \{1, 2, 4\}, c_2 = \{1, 3\}, c_3 = \{2, 3\}, c_4 = \{3, 5\}\} .$$

In Zeile 3 werde jeweils die Menge c_i mit dem kleinsten Index i unter allen Mengen in $\mathcal{C}$ gewählt.

Lösung: $X = \{1, 2, 3, 4, 5\}$

(b) Zeigen Sie, dass Algorithmus 3 ein 3-Approximationsalgorithmus ist, d.h. dass

$$\mathcal{A}(I) \leq 3 \cdot OPT(I)$$

gilt. Hierbei sei I eine beliebige Instanz des Problems, $\mathcal{A}(I)$ die Kardinalität des von Algorithmus 3 zurückgegebenen Hitting Sets sowie $OPT(I)$ die Kardinalität einer optimalen Lösung.

Hinweis: Betrachten Sie eine (feste) optimale Lösung X^*. Zeigen Sie, dass jede in Zeile 3 ausgewählte Menge A mindestens ein Element $z \in X^* \setminus X$ enthält.

Lösung: Wir zeigen zuerst den Hinweis: Da X^* ein Hitting Set ist, gilt nach Definition $X^* \cap A \neq \emptyset$. Sei also $z \in X^* \cap A$ beliebig, aber fest. Wir nehmen nun an, dass $z \in X$ gilt. Dann wurde die Schleife mindestens einmal durchlaufen, da X als leere Menge initialisiert wurde. Dies ist aber ein Widerspruch, da in diesem Fall A in Zeile 5 aus $\mathcal{C}$ entfernt worden wäre. Also ist $z \in A \cap (X^* \setminus X)$ wie behauptet.

In jedem Durchlauf von Zeile 4 werden wegen $|A| \leq 3$ höchstens drei Elemente zu X hinzugefügt, von denen mindestens eines in der optimalen Lösung X^*, nicht aber in X enthalten ist. Für jedes Element $x^* \in X^*$ fügen wir also höchstens einmal je drei Elemente zu X hinzu. Daher gilt $\mathcal{A}(I) = |X| \leq 3 \cdot |X^*| = 3 \cdot OPT(I)$.

Problem 8: Randomisierte Algorithmen 2 + 1 + 3 = 6 Punkte

Ein d-regulärer Graph $G = (V, E)$ ist ein Graph mit der Eigenschaft, dass jeder Knoten $v \in V$ *genau* d Nachbarn hat.

Algorithmus 4 : Randomisierter Algorithmus

Eingabe : d-regulärer Graph $G = (V, E)$, Wahrscheinlichkeit $p \in [0, 1]$
Ausgabe : Menge S

1 $S \leftarrow \emptyset$
2 **Für** *jedes* $v \in V$
3 $\quad x_v \leftarrow$ blau
4 $\quad x_v \leftarrow$ rot mit Wahrscheinlichkeit p
5 **Für** *jedes* $v \in V$
6 $\quad$ **Wenn** $x_v = rot$ *und* $x_w = blau$ *für alle* $w \in V$ *mit* $\{v, w\} \in E$
7 $\quad\quad S \leftarrow S \cup \{v\}$

(a) Sei S eine Ausgabe von Algorithmus 4. Zeigen Sie: Für alle $\{v, w\} \in E$ gilt $|\{v, w\} \cap S| \leq 1$, d.h. für jede Kante ist höchstens einer der beiden Endknoten in S.

Lösung: Offenbar ist $|\{v, w\} \cap S| \leq 2$. Sei also $|\{v, w\} \cap S| = 2$. In S befinden sich ausschließlich Knoten z mit $x_z =$ rot (Zeile 6). Damit gilt $x_v =$ rot und $x_w =$ rot. Andererseits müssen alle Nachbarn z von v die Bedingung $x_z =$ blau erfüllen (Zeile 6). Dies gilt auch für w, da w ein Nachbar von v ist, d.h. $x_w =$ blau. Dies ist ein Widerspruch und es gilt die Annahme.

(b) Geben Sie die Wahrscheinlichkeit $\Pr(v \in S)$ dafür an, dass ein gegebener Knoten v in S liegt.

Lösung:

$$\Pr(v \in S) = p \cdot (1 - p)^d$$

(c) Geben Sie den Erwartungswert von $|S|$ in Abhängigkeit von d, p und $n := |V|$ an.

Hinweis: Betrachten Sie die Zufallsvariablen $Y_v = \begin{cases} 1, & v \in S \\ 0, & \text{sonst} \end{cases}$.

Lösung:

$$\mathbb{E}(|S|) = \mathbb{E}(\sum_{v \in V} Y_v) \tag{7}$$

$$= \sum_{v \in V} \mathbb{E} Y_v \tag{8}$$

$$= \sum_{v \in V} p \cdot (1 - p)^d \tag{9}$$

$$= n \cdot p \cdot (1 - p)^d \tag{10}$$

Problem 9: $12 \times 1 = 12$ 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.

Aussage	Wahr	Falsch
Ein einzelner Aufruf der Operation Find einer Union-Find-Datenstruktur auf n Elementen benötigt maximal $\mathcal{O}(G(n))$ Zeit.		X
Seien T_1 und T_2 zwei Spannbäume eines zusammenhängenden Graphen G mit Kantenmengen $E(T_1)$ bzw. $E(T_2)$. Dann induziert die Symmetrische Differenz von $E(T_1)$ und $E(T_2)$ wieder einen Spannbaum.		X
Gegeben sei ein Graph $G = (V, E)$. Eine Menge $M \subseteq V$ heißt unabhängig, falls für je zwei $x, y \in M$ gilt: $\{x, y\} \notin E$. Die Menge aller unabhängigen Mengen eines Graphen ist eine Unabhängigkeitssystem, aber kein Matroid.	X	
Die Menge der Fundamentalkreise eines ungewichteten Graphen G mit Spannbaum T ist stets eine Kreisbasis minimalen Gewichts.		X
Sei $\mathcal{I}$ ein ganzzahliges lineares Programm (ILP). Dann besitzt $\mathcal{I}$ nur endlich viele zulässige Lösungen.		X
Die Simplex-Methode ist ein Algorithmus zum Lösen von linearen Programmen mit exponentieller worst-case Laufzeit.	X	
Sei G ein Graph und C ein Vertex Cover von G der Größe k. Dann enthält C alle Knoten mit k oder weniger Nachbarn.		X
Sei $\mathcal{A}$ ein randomisierter Algorithmus. Dann gibt $\mathcal{A}$ für jede Eingabe mindestens zwei verschiedene Ausgaben A_1 und A_2 mit echt positiver Wahrscheinlichkeit aus.		X
Sei Π ein Problem mit Parameter k und sei $\mathcal{A}$ ein Lösungsalgorithmus für Π mit Laufzeit in $O(2^{2^k} \cdot n^5)$, wobei n die Problemgröße bezeichnet. Dann ist Π fixed parameter tractable.	X	
Ein BROADCAST kann im EREW-PRAM Modell schneller ausgeführt werden als im CRCW-Modell.		X
Sei $\mathcal{A}$ ein 2-approximativer Algorithmus für ein Optimierungsproblem Π. Dann ist die Laufzeit von $\mathcal{A}$ höchstens doppelt so groß wie die Laufzeit eines optimalen Algorithmus.		X
Jedes PAS ist ein APAS.	X	