

1. Klausur zur Vorlesung
Algorithmentechnik
Wintersemester 2009/2010

Lösung!

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 | 1 | 2 | 2 | - | 5 | | | | - | |
| 2 | 1 | 2 | 3 | - | 6 | | | | - | |
| 3 | 3 | 1 | - | - | 4 | | | - | - | |
| 4 | 4 | 2 | - | - | 6 | | | - | - | |
| 5 | 1 | 2 | 2 | - | 5 | | | | - | |
| 6 | 3 | 2 | 3 | 2 | 10 | | | | | |
| 7 | 3 | 4 | - | - | 7 | | | - | - | |
| 8 | 2 | 4 | - | - | 6 | | | - | - | |
| 9 | 11x1 | | | | 11 | | | | | |
| Σ | | | | | 60 | | | | | |

Problem 1: Union-Find

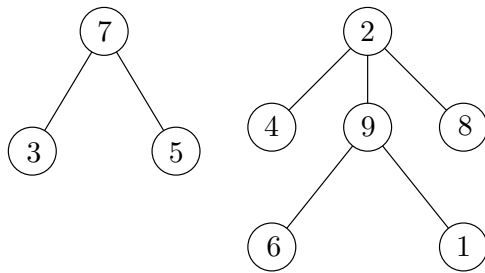
1 + 2 + 2 = 5 Punkte

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

- (a) Gegeben sei folgendes Array A , das den Zustand einer Union-Find-Datenstruktur mit 9 Elementen repräsentiert. Zeichnen Sie die in A gegebene Datenstruktur in der Baum-Repräsentation (wie in der Vorlesung vorgestellt).

| x | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|--------|---|----|---|---|---|---|----|---|---|
| Vor(x) | 9 | -6 | 7 | 2 | 7 | 9 | -3 | 2 | 2 |

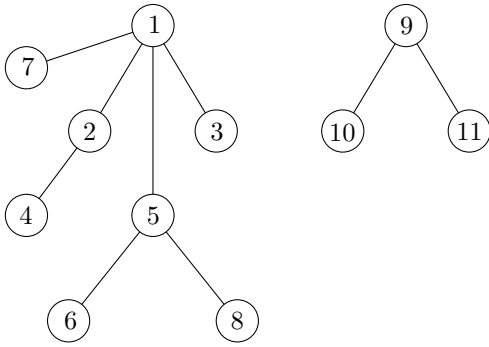
Lösung:



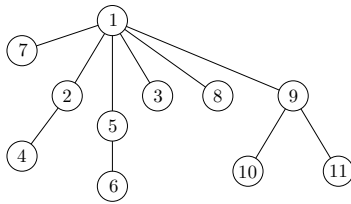
(b) Gegeben sei folgende Union-Find-Datenstruktur in der Baum-Repräsentation sowie die Operationen

- (i) UNION(FIND(8),FIND(10)) und
- (ii) FIND(11).

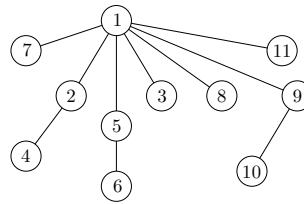
Zeichnen Sie den Zustand der Datenstruktur (als Baum-Repräsentation) sowohl nach Ausführung der Operation (i) als auch nach **anschließender** Ausführung der Operation (ii).



Lösung:



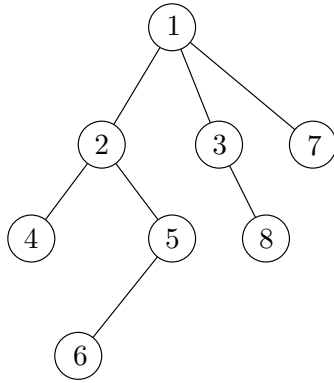
nach Operation (i)



nach Operation (ii)

- (c) Betrachten Sie den unten 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*.

MAKESET(1), ..., MAKESET(8)



Lösung:

MAKESET(1), ..., MAKESET(8)

UNION(5,6), UNION(2,4), UNION(2,5),

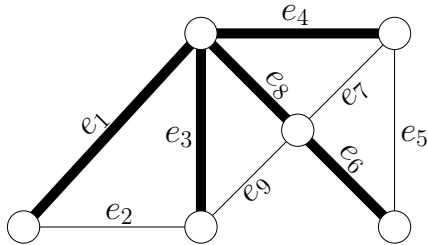
UNION(3,8), UNION(1,7), UNION(1,3),

UNION(1,2)

Problem 2: Kreisbasen

1 + 2 + 3 = 6 Punkte

- (a) Untenstehende Abbildung zeigt den Graphen $G^* := (V^*, E^*)$. Gegeben sei weiterhin der aufspannende Baum $T^* = \{e_1, e_3, e_4, e_6, e_8\}$ (fett eingezeichnet). Geben Sie die durch T^* induzierte Fundamentalbasis an. Stellen Sie die Fundamentalkreise als Teilmengen von E^* dar.



Lösung: $C_2 := \{e_1, e_2, e_3\}$, $C_5 := \{e_4, e_5, e_6, e_8\}$, $C_7 := \{e_4, e_7, e_8\}$, $C_9 := \{e_3, e_8, e_9\}$

- (b) Gegeben sei ein Graph $G := (V, E)$. Weiter sei $T := (V, E_T)$ ein aufspannender Baum von G , und $e \in E_T$ sei eine **Baumkante**. Die Kante e sei in zwei Fundamentalkreisen C_1 und C_2 bzgl. T enthalten.

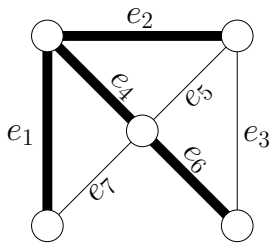
Sei $T' = (V, E'_T)$ ein weiterer aufspannender Baum von G , der dieselbe Fundamentalbasis induziert wie T . Zeigen Sie, dass e in E'_T enthalten sein muss.

Lösung: Annahme: $e \notin E'_T$.

Dann induziert e als Nichtbaumkante bzgl. T' einen Fundamentalkreis C_e . Damit ist C_e aber der einzige Fundamentalkreis bzgl. T' , in dem e vorkommt. Widerspruch!

- (c) Untenstehende Abbildung zeigt den Graphen $\hat{G} := (\hat{V}, \hat{E})$. Gegeben sei weiterhin der aufspannende Baum $\hat{T} = \{e_1, e_2, e_4, e_6\}$ (fett eingezeichnet) sowie die zugehörige Fundamentalebasis mit $C_3 := \{e_2, e_3, e_4, e_6\}$, $C_5 := \{e_2, e_4, e_5\}$ und $C_7 := \{e_1, e_4, e_7\}$.

Geben Sie alle weiteren aufspannenden Bäume von \hat{G} an, die dieselbe Fundamentalebasis induzieren wie \hat{T} . Begründen Sie, weshalb die von Ihnen angegebene Baummenge maximal ist.



Lösung: Mit (b) müssen die Kanten e_2 und e_4 in allen Bäumen vorkommen, da sie in mehreren Fundamentalkreisen liegen (z.B. in C_3 und C_5). Damit kann e_5 in keinem Baum vorkommen, da sonst ein Kreis entstünde. Um C_3 zu erhalten muss entweder e_3 oder e_6 vorkommen. Um C_7 zu erhalten muss entweder e_1 oder e_7 vorkommen.

(Alternative Begründung für entweder e_3 oder e_6 und entweder e_1 oder e_7 :

- Mindestens jeweils eine Kante jedes Paares, da sonst kein Spannbaum entsteht.
- Maximal jeweils eine Kante jedes Paares, da sonst ein Kreis entsteht.

Damit ergeben sich genau drei weitere Bäume: $\hat{T}_1 := \{e_1, e_2, e_3, e_4\}$, $\hat{T}_2 := \{e_2, e_4, e_6, e_7\}$, $\hat{T}_3 := \{e_2, e_3, e_4, e_7\}$

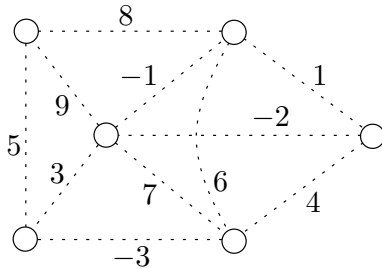
Problem 3: Maybe-MST

3 + 1 = 4 Punkte

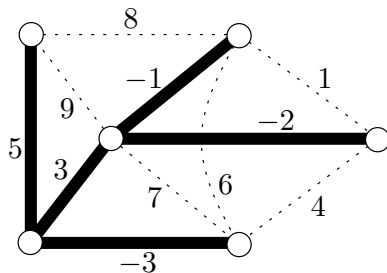
Gegeben sei folgender Algorithmus MAYBE-MST. Die Eingabe sei ein gewichteter Graph $G = (V, E, c)$ mit Gewichtungsfunktion $c : E \rightarrow \mathbb{R}$. Desweiteren bezeichne $\min(v) \subseteq E$ die Menge der minimal gewichteten Kanten unter allen zu $v \in V$ inzidenten Kanten.

Algorithmus 1 : MAYBE-MST**Eingabe :** Graph $G = (V, E, c)$ **Ausgabe :** Kantenmenge T 1 $T \leftarrow \emptyset$ 2 **für alle** $v \in V$ **tue**3 $T \leftarrow T \cup \min(v)$ 4 Sortiere $E \setminus T$ in nicht absteigender Reihenfolge5 **Für** sortierte Kanten, wende nacheinander an:6 **Wenn** Endknoten der Kante in verschiedenen Zusammenhangskomponenten von T liegen7 \lfloor Füge Kante zu T hinzu8 Return T

- (a) Wenden Sie MAYBE-MST auf den unten abgebildeten Graphen G^* an. Zeichnen Sie die Kanten in T fett durchgezogen. Berechnet MAYBE-MST für G^* einen minimalen aufspannenden Baum?



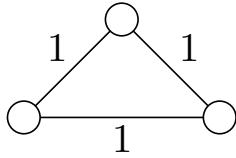
Lösung: Ja, MAYBE-MST berechnet auf G^* einen minimalen aufspannenden Baum.



Kruskals Algorithmus liefert den gleichen Baum, damit ist die minimalität gezeigt.

- (b) Zeichnen Sie einen nicht-leeren, zusammenhängenden, gewichteten Eingabegraphen, für den MAYBE-MST **keinen** minimalen aufspannenden Baum berechnet.

Lösung: Hier ist T bereits nach Abarbeitung der Schleife in Zeile 2 kein Baum mehr.



Problem 4: Matroide

4 + 2 = 6 Punkte

Sei $G = (V, E)$ ein einfacher, gerichteter Graph. Für jeden Knoten $v \in V$ sei durch $k_v \in \mathbb{Z}^+$ eine positive, ganze Zahl gegeben.

Wir definieren ein Mengensystem (M, \mathcal{U}) durch $M := E$ und

$$\mathcal{U} := \{I \subseteq E \mid \text{indeg}_I(v) \leq k_v \text{ für alle } v \in V\}.$$

Dabei bezeichne $\text{indeg}_I : V \rightarrow \mathbb{N}_0$ den *Eingangsgrad* eines Knotens bezüglich der Kantenmenge I , das heißt $\text{indeg}_I(v) := |\{u \in V \mid (u, v) \in I\}|$.

(a) Zeigen Sie, dass (M, \mathcal{U}) ein Matroid ist.

Lösung: Wir weisen die Matroideigenschaften nach.

- Zu Zeigen: $\emptyset \in \mathcal{U}$:

Bezüglich der leeren Menge \emptyset gilt $\text{indeg}_{\emptyset}(v) = 0 \leq k_v$ für alle $v \in V$. Damit ist $\emptyset \in \mathcal{U}$.

- Zu Zeigen: $I \in \mathcal{U}$ und $I' \subseteq I$ dann ist auch $I' \in \mathcal{U}$:

Ist $I \in \mathcal{U}$ so gilt für alle $v \in V$ dass $\text{indeg}_I(v) \leq k_v$. Wegen $I' \subseteq I$ gilt nun dass $\text{indeg}_{I'}(v) \leq \text{indeg}_I(v) \leq k_v$ weiterhin erfüllt ist, das heißt $I' \in \mathcal{U}$.

- Zu Zeigen: $I, J \in \mathcal{U}$ mit $|I| < |J|$, dann gibt es eine Kante $e \in J \setminus I$, so dass $I \cup \{e\} \in \mathcal{U}$:

Wegen $|I| < |J|$ gibt es einen Knoten $v \in V$ für den gilt $\text{indeg}_I(v) < \text{indeg}_J(v) \leq k_v$. Das heißt, es gibt eine Kante $e \in J$ mit $e = (u, v)$, die zu $\text{indeg}_J(v)$ beiträgt, aber nicht in I ist. Somit ist $I' := I \cup \{e\}$ ebenfalls in \mathcal{U} , da gilt $\text{indeg}_I(v) + 1 = \text{indeg}_{I'}(v) \leq \text{indeg}_J(v) \leq k_v$.

Somit ist (M, \mathcal{U}) ein Matroid.

Sei nun $G = (V, E)$ ein einfacher, *ungerichteter* Graph. Analog zu (a) sei das Mengensystem (M, \mathcal{U}) definiert durch $M := E$ und

$$\mathcal{U} := \{I \subseteq E \mid \text{deg}_I(v) \leq k_v \text{ für alle } v \in V\}.$$

Dabei ist $\text{deg}_I : V \rightarrow \mathbb{N}_0$ der *Grad* eines Knotens bezüglich der Kantenmenge I , das heißt $\text{deg}_I(v) := |\{u \in V \mid \{u, v\} \in I\}|$.

(b) Zeigen Sie durch Angabe eines Gegenbeispiels, dass (M, \mathcal{U}) im Allgemeinen kein Matroid ist.

Lösung: Dass diese Struktur im Gegensatz zu Aufgabe (a) kein Matroid ist, liegt daran, dass *eine* Kante Einfluss auf *zwei* Knoten hat. Folgender Graph liefert ein Gegenbeispiel:

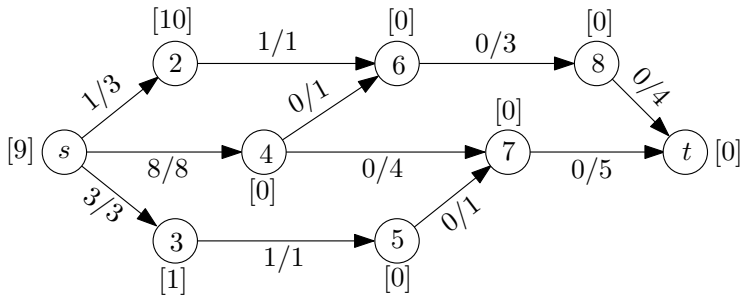


Die Zahl in jedem Knoten v stehe für den Wert k_v . Dann kann aus der blau gestrichelten Menge J keine Kante $e \in J$ zu der rot gepunkteten Menge I (mit $|I| < |J|$) hinzugenommen werden, ohne die Bedingung $\text{deg}_{I \cup \{e\}}(v) \leq k_v$ an einem der Knoten zu verletzen. Somit ist die Matroideigenschaft im Allgemeinen nicht gegeben.

Problem 5: Goldberg-Tarjan-Fluss-Algorithmus

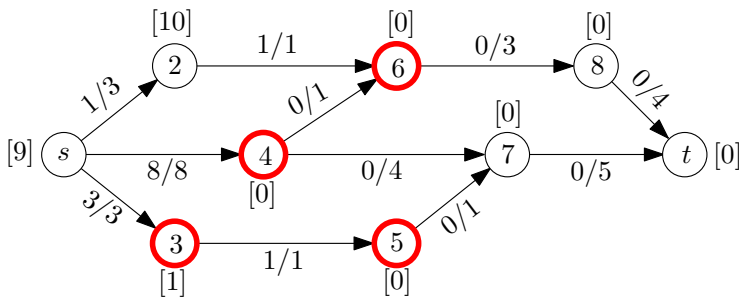
1 + 2 + 2 = 5 Punkte

Untenstehendes Netzwerk $(D; s; t; c)$ zeigt einen Zwischenschritt des Goldberg-Tarjan-Algorithmus. Die Distanzlabel $dist(v)$ der Knoten (aus der Prozedur RELABEL im Skript) sind in eckigen Klammern über den Knoten notiert, die Kanten sind im Format $f(e)/c(e)$ beschriftet. Dabei bezeichnet $c(e)$ die Kantenkapazität und $f(e)$ den Fluss auf der Kante e .



- (a) Markieren Sie in obiger Abbildung alle aktuell aktiven Knoten.

Lösung:

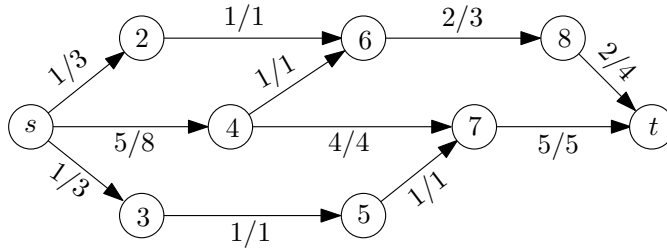


- (b) Ausgehend von obigem Zustand, geben Sie eine mögliche Folge von PUSH- und RELABEL-Aufrufen an, nach deren Ausführung Knoten 4 erstmals nicht mehr aktiv ist (Hinweis: Solange Knoten 4 aktiv ist, wählen Sie diesen als aktuellen aktiven Knoten).

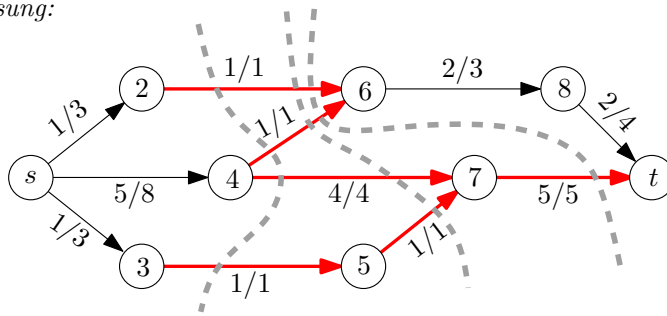
Lösung:

RELABEL(4) auf $dist(4) = 1$
 PUSH von 4 nach 6 den Wert 1
 PUSH von 4 nach 7 den Wert 4
 RELABEL(4) auf $dist(4) = 10$
 PUSH von 4 nach s den Wert 3

- (c) Untenstehende Abbildung zeigt einen maximalen s - t -Fluss auf dem Netzwerk. Zeichnen Sie alle minimalen s - t -Schnitte ein (Schnittgewicht bzgl. Kantenkapazität).



Lösung:



Problem 6: Maximum Matching

3 + 2 + 3 + 2 = 10 Punkte

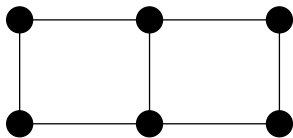
In einem einfachen, ungewichteten Graphen $G = (V, E)$ ist ein *Matching* eine Kantenmenge $M \subseteq E$, so dass keine zwei Kanten aus M zu einem gemeinsamen Knoten inzident sind. Ein Matching M heißt

- **inklusionsmaximal**, wenn es keine echte Teilmenge eines anderen Matchings M' ist, d.h. es existiert kein Matching M' , so dass $M \subsetneq M'$.
- **kardinalitätsmaximal**, wenn es kein Matching M' mit echt größerer Kantenzahl gibt, d.h. es existiert kein Matching M' , so dass $|M| < |M'|$.

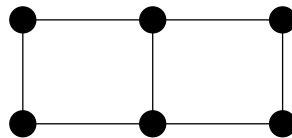
(a) Zeichnen Sie in die untenstehenden Graphen Folgendes ein:

- in (i) ein **kardinalitätsmaximales** Matching
- in (ii) ein **inklusionsmaximales** aber **kein kardinalitätsmaximales** Matching
- in (iii) ein nicht-leeres, **nicht inklusionsmaximales** Matching

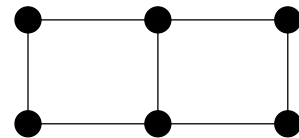
Zeichnen Sie hierzu die Kanten der Matchings fett ein.



(i) **kardinalitätsmaximal**.

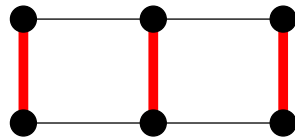


(ii) **inklusionsmaximal**,
nicht kardinalitätsmaximal.

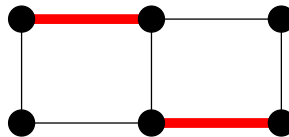


(iii) nicht leer,
nicht inklusionsmaximal.

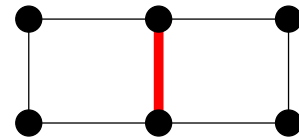
Lösung:



(i) **kardinalitätsmaximal**.



(ii) **inklusionsmaximal**,
nicht kardinalitätsmaximal.



(iii) nicht leer,
nicht inklusionsmaximal.

Eine Knotenmenge $V' \subseteq V$ heißt *Überdeckung* der Kantenmenge E , wenn für alle $e \in E$ gilt:
 $\exists u \in V'$ mit u inzident zu e .

- (b) Sei M ein beliebiges Matching in $G = (V, E)$.
 Sei V' eine beliebige Überdeckung von E .

Zeigen Sie: $|M| \leq |V'|$,

wobei $|M|$ die Anzahl der Kanten in M und $|V'|$ die Anzahl der Knoten in V' bezeichnet.

Lösung: Jeder Knoten in V kann maximal eine Kante aus M überdecken. Mit $M \subseteq E$ braucht man also mindestens $|M|$ Knoten um M und damit ganz E zu überdecken.

- (c) Sei M_{\max} ein **inklusions**maximales Matching in $G = (V, E)$.
 Sei V'_{\min} eine Überdeckung von E mit minimaler Kardinalität.

Zeigen Sie: $|V'_{\min}| \leq 2|M_{\max}|$,

wobei $|V'_{\min}|$ die Anzahl der Knoten in V'_{\min} und $|M_{\max}|$ die Anzahl der Kanten in M_{\max} bezeichnet.

Lösung: Jede Kante $\notin M_{\max}$ hat mindestens einen Knoten mit einer Kante $\in M_{\max}$ gemeinsam, da sonst das Matching erweiterbar wäre. Damit kann jede Kante $\notin M_{\max}$ von einem Knoten überdeckt werden, der inzident zu einer Kante $\in M_{\max}$ ist, was für die jeweilige Kante $\in M_{\max}$ ohnehin gilt. Insgesamt gibt es $2|M_{\max}|$ Knoten, die zu einer Kante $\in M_{\max}$ inzident sind. Diese reichen aus um ganz E zu überdecken.

- (d) Sei I eine Instanz des Matchingproblems.
 Sei M_M ein **kardinalitäts**maximales Matching bzgl. I .
 Sei M_{\max} ein **inklusions**maximales Matching bzgl. I .

Zeigen Sie: $\text{OPT}(I) \leq 2 \cdot \mathcal{A}(I)$,

mit $\text{OPT}(I) := |M_M|$, $\mathcal{A}(I) := |M_{\max}|$.

(Hinweis: Man kann die Teilaufgaben (b) und (c) nutzen.)

Lösung: Zu zeigen: $|M_M| \leq 2|M_{\max}|$.

Sei also \hat{V} eine Überdeckung von E mit minimaler Kardinalität.

Mit (b) folgt $|M_M| \leq |\hat{V}|$.

Mit (c) folgt $|\hat{V}| \leq 2|M_{\max}|$.

Insgesamt folgt $|M_M| \leq |\hat{V}| \leq 2|M_{\max}|$.

Problem 7: Lineares Programm für kürzeste Wege

3 + 4 = 7 Punkte

Gegeben sei ein einfacher, gerichteter, gewichteter Graph $G = (V, E, w)$ mit einer *echt-positiven* Gewichtsfunktion $w : E \rightarrow \mathbb{R}^+$ und zwei Knoten $s, t \in V$ mit $s \neq t$.

Ein s - t -Weg ist ein Weg mit Startknoten s und Endknoten t .

Ein *kürzester* s - t -Weg ist ein s - t -Weg mit minimalem Gewicht unter allen s - t -Wegen.

- (a) Gegeben sei ein kürzester s - t -Weg $P = (s = v_1, v_2, \dots, v_{k-1}, v_k = t)$. Weiterhin sei für alle Kanten $(u, v) \in E$

$$x_{u,v} := \begin{cases} 1, & (u, v) \in P \\ 0, & \text{sonst.} \end{cases}$$

Zeigen Sie, dass für jeden Knoten $v \in V$ mit $v \neq s, t$ gilt:

$$\sum_{(u,v) \in E} x_{u,v} - \sum_{(v,u) \in E} x_{v,u} = 0$$

Lösung: Da P ein kürzester Weg ist und G echt positiv gewichtet ist, muss P einfach sein. Wir unterscheiden zwei Fälle:

- v liegt auf P : Dann hat v einen Vorgänger u und einen Nachfolger w auf P . Da P einfach ist, sind (u, v) und (v, w) die einzigen Kanten in P , die inzident zu v sind. Damit gilt $x_{u,v} = x_{v,w} = 1$ und $x_e = 0$ für alle übrigen Kanten die inzident zu v sind; Also $x_{u,v} - x_{v,w} = 0$.
 - v liegt nicht auf P : Dann gilt für alle Kanten e inzident zu v , dass sie nicht auf P liegen, also $x_e = 0$.
- (b) Geben Sie ein (ganzzahliges) lineares Programm an, dessen Optimallösung einen kürzesten s - t -Weg P liefert. Wählen Sie für jede Kante $(u, v) \in E$ als Variable $x_{u,v} \in \{0, 1\}$ mit der gleichen Bedeutung wie in Aufgabe (a).

Lösung: Zielfunktion:

$$\text{minimiere } \sum_{(u,v) \in E} x_{u,v} \cdot w(u, v)$$

(Finde den Weg (als Kantenmenge) mit minimalem Gesamtgewicht.)

Nebenbedingungen:

$$\text{Für alle } (u, v) \in E : \quad x_{u,v} \in \{0, 1\}$$

$$\text{Für alle } v \in V : \quad \sum_{(u,v) \in E} x_{u,v} - \sum_{(v,u) \in E} x_{v,u} = \begin{cases} -1 & \text{falls } v = s \\ 1 & \text{falls } v = t \\ 0 & \text{sonst} \end{cases}$$

Die Variable $x_{u,v}$ modelliert, ob eine Kante Teil eines kürzesten s - t -Weges P ist. Für jeden Knoten $v \neq s, t$ gilt, dass die Differenz aus der Anzahl eingehender Kanten zu v und Anzahl ausgehender Kanten von v immer 0 ergibt. Da der Weg bei s beginnt und bei t endet, ist diese Differenz bei s gerade -1 und bei t ist sie 1 .

Problem 8: Kernbildung für 3-HITTING SET

2 + 4 = 6 Punkte

Das 3-HITTING SET-Problem ist wie folgt definiert.

Gegeben: Eine Menge V und eine Menge $\Phi \subseteq 2^V$ von Teilmengen von V .
Dabei hat jede Menge in Φ höchstens 3 Elemente.
Ein Parameter k .

Gesucht: Eine Teilmenge $H \subseteq V$ mit $|H| \leq k$,
so dass H mindestens ein Element aus jeder der Mengen aus Φ enthält.

(a) Betrachten Sie folgende Instanz:

$V = \{1, 2, 3, 4, 5\}$, $\Phi = \{\{1, 2, 3\}, \{2, 5\}, \{3, 4, 5\}\}$, $k = 2$.

Geben Sie zwei verschiedene 3-Hitting Sets aus jeweils maximal 2 Elementen an.

Lösung: $\{2, 5\}$ und $\{2, 3\}$

(b) Sei $x, y \in V$ mit $x \neq y$,

und sei $\Phi_{x,y} \subseteq \Phi$ die Menge aller Tripel, die x und y enthalten.

Außerdem sei $|\Phi_{x,y}| > k$.

Zeigen Sie: Jedes 3-Hitting Set H für Φ mit $|H| \leq k$
ist auch ein 3-Hitting Set für $\Phi' := (\Phi \setminus \Phi_{x,y}) \cup \{\{x, y\}\}$.

Lösung: Sei H ein Hitting-Set für Φ' mit $|H| \leq k$. $\Phi' \setminus \{\{x, y\}\}$ ist eine Teilmenge von Φ und wird daher von H überdeckt. Mit $|\Phi_{x,y}| > k$ kann $\Phi_{x,y}$ nicht mit $|\Phi_{x,y}|$ verschiedenen Elementen in H überdeckt werden (da H sonst zu groß würde). Damit muss H mindestens x oder y beinhalten und überdeckt so auch die Menge $\{x, y\}$.

Problem 9: $11 \times 1 = 11$ 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.

Auf ein Fluss-Netzwerk $(D; s; t; c)$ mit ganzzahligen Kapazitäten wird der FORD-FULKERSON-Algorithmus angewendet. Der so berechnete maximale s - t -Fluss ist dann ebenfalls auf jeder Kante ganzzahlig.

| | |
|-------------------------------------|--------------------------|
| <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Wahr | Falsch |

Jedes APAS ist ein FPAS.

| | |
|--------------------------|-------------------------------------|
| <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Wahr | Falsch |

Die Klasse \mathcal{NC} (Nick's Class) ist die Klasse der Probleme, die durch einen parallelen Algorithmus mit polylogarithmischer Laufzeit und polynomieller Prozessorzahl gelöst werden kann.

| | |
|-------------------------------------|--------------------------|
| <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Wahr | Falsch |

Das MAX-2-SAT-Problem ist \mathcal{NP} -schwer.

| | |
|-------------------------------------|--------------------------|
| <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Wahr | Falsch |

Jede Fundamentalbasis bezüglich eines MST ist eine minimale Kreisbasis.

| | |
|--------------------------|-------------------------------------|
| <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Wahr | Falsch |

Für ein Problem mit der Rekursionsgleichung $T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + n$ gilt $T(n) \in \Theta(n \cdot \log n)$.

| | |
|-------------------------------------|--------------------------|
| <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Wahr | Falsch |

Sei P ein lineares Programm mit optimalem Lösungsvektor x^* , und D sei das zugehörige duale Programm mit optimalem Lösungsvektor y^* , so gilt $x^* = y^*$.

| | |
|--------------------------|-------------------------------------|
| <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Wahr | Falsch |

In jeder Phase des Stoer & Wagner Schnitt-Algorithmus kann der Startknoten beliebig gewählt werden, ohne dass die Korrektheit des Algorithmus dadurch beeinflusst wird.

| | |
|-------------------------------------|--------------------------|
| <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Wahr | Falsch |

Sei G ein Graph, C ein Vertex Cover und D ein Dominating Set von G . Dann gilt $|C| \leq |D|$.

| | |
|--------------------------|-------------------------------------|
| <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Wahr | Falsch |

Sei \mathcal{A} ein 2-Approximationsalgorithmus für ein Maximierungsproblem, dann gilt $\mathcal{A}(I) \geq \frac{1}{2} \text{OPT}(I)$. Dabei beschreibt I eine Problem Instanz.

| | |
|-------------------------------------|--------------------------|
| <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Wahr | Falsch |

Sei G ein Graph und T ein aufspannender Baum von G . Dann trennt jeder nicht-triviale Schnitt in G ein in T adjazentes Knotenpaar.

| | |
|-------------------------------------|--------------------------|
| <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Wahr | Falsch |