TU Wien
Institut für Logic and Computation
Algorithms and Complexity Group

186.866 Algorithmen und Datenstrukturen VU 8.0

2. Test, 2020S

26. Juni 2020

Gruppe A

Hörsaal: ______ Platz: ______

Machen Sie die folgenden Angaben in deutlicher **Blockschrift**:

Nachname: ______ Vorname: ______

Matrikelnummer: ______ Unterschrift: ______

Sie dürfen die Lösungen nur auf die Angabeblätter schreiben, die Sie von der Aufsicht erhalten. Es ist nicht zulässig, eventuell mitgebrachtes eigenes Papier zu verwenden. Benutzen Sie dokumentenechte Schreibgeräte (keine Bleistifte!).

Die Verwendung von Taschenrechnern, Mobiltelefonen, Tablets, Digitalkameras, Skripten, Büchern, Mitschriften, Ausarbeitungen oder vergleichbaren Hilfsmitteln ist unzulässig.

Kennzeichnen Sie bei Ankreuzfragen eindeutig, welche Kästchen Sie kreuzen. Streichen Sie Passagen, die nicht gewertet werden sollen, deutlich durch. Unleserliche Antworten werden nicht gewertet.

	A1	A2	A3	A4	A5	Summe
Erreichbare Punkte:	20	20	20	20	20	100
Erreichte Punkte:						

Viel Erfolg!

Aufgabe A1: P-NP und Polynomialzeitreduktionen (20 Punkte)

a) (12 Punkte) Seien A, B, C Ja/Nein-Probleme und n die Eingabegröße. Weiters seien A und B in NP. Nehmen Sie an, es gibt

- eine Reduktion von A nach B in Zeit $O(n^2)$,
- eine Reduktion von B nach SAT in Zeit $O(n \cdot \log n)$,
- eine Reduktion von SAT nach A in Zeit $O(n^3)$,
- eine Reduktion von C nach A in Zeit $O(n!)$.

(i) Kreuzen Sie in den folgenden Tabellen jeweils die zutreffenden Felder an:

(je korrekter Zeile 1 Punkt, keine Minuspunkte)

A ist ...	Ja	Nein	Keine Aussage möglich
... NP-schwer			
... NP-vollständig			

B ist ...	Ja	Nein	Keine Aussage möglich
... NP-schwer			
... NP-vollständig			

C ist ...	Ja	Nein	Keine Aussage möglich
... NP-schwer			
... NP-vollständig			

(ii) Für welche der Probleme A, B, C und SAT würde ein polynomieller Algorithmus zeigen, dass P=NP gilt?

(iii) Nehmen Sie nun an, SAT kann für Eingaben der Größe n in Zeit $O(n)$ gelöst werden. Welche engste obere Schranke können Sie aus den gegebenen Informationen für die Worst-Case Laufzeit eines optimalen Algorithmus für A schließen?

b) (8 Punkte) Im Folgenden sind zwei Probleme mit verschiedenen Zertifikaten gegeben. Welche Zertifikate sind (gemeinsam mit einem geeigneten Zertifizierer) geeignet, um zu zeigen, dass das gegebene Problem in NP ist? Kreuzen Sie Zutreffendes an. Nehmen Sie hierfür P $\neq$ NP an.

Es kann mehr als eine richtige Antwort geben.

(je Unteraufgabe: alles korrekt: 4 Punkte, ein Fehler: 2 Punkte, sonst / kein Kreuz: 0 Punkte)

i) **Problem:** Gegeben sei ein Graph G mit n Knoten und m Kanten und eine natürliche Zahl k. Gibt es ein Independent Set von G mit k oder mehr Knoten?

Zertifikat:

- ☐ Ein Independent Set der Größe k.
- ☐ Ein leerer String.
- ☐ Ein Vertex Cover mit $n - k$ Knoten.
- ☐ Keines der Zertifikate ist geeignet.

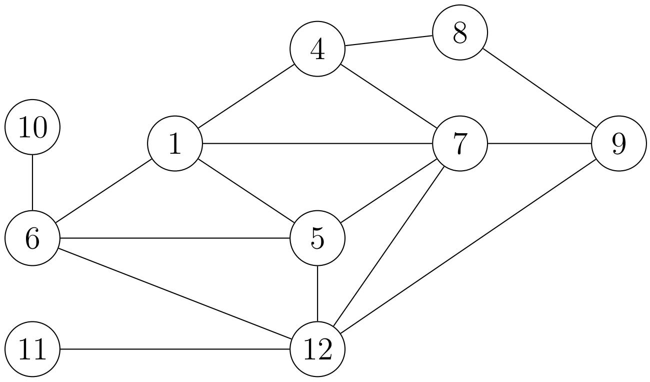
ii) **Problem:** Gegeben sei ein Graph G. Hat das *kleinste* Vertex Cover von G *genau* k Knoten?

Zertifikat:

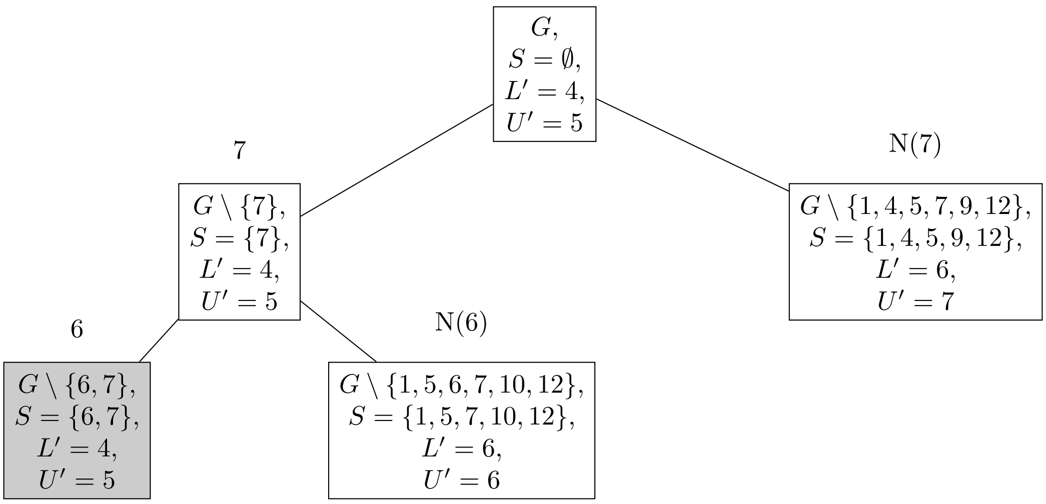
- ☐ Ein Vertex Cover mit weniger als k Knoten.
- ☐ Ein Vertex Cover mit genau k Knoten.
- ☐ Die Größe l des kleinsten Vertex Covers.
- ☐ Keines der Zertifikate ist geeignet.

Aufgabe A2: Branch and Bound **(20 Punkte)**

Auf dem folgenden Graphen soll mittels des Branch and Bound Algorithmus aus der Vorlesung ein minimales Vertex Cover gefunden werden.



Betrachten Sie die folgende Skizze eines partiellen Ablaufs des Branch and Bound Algorithmus. In der Skizze geben die ersten beiden Zeilen das aktuelle Teilproblem an. In der ersten Zeile wird der Graph des aktuellen Teilproblems angegeben, in der zweiten Zeile die aktuelle partielle Lösung S. Weiter steht L' für die lokale untere Schranke und U' für die lokale obere Schranke.



a) (3 Punkte) Wird bei dem Algorithmus eine globale obere Schranke oder eine globale untere Schranke verwaltet? Geben Sie den Wert der Schranke zum Zeitpunkt, wie auf der vorigen Seite abgebildet, an.

b) (3 Punkte) Kreuzen Sie an, welche Blätter des Baums in der Skizze auf der vorigen Seite nicht mehr weiter aufgespalten werden müssen.

(alles korrekt: 3 Punkte, ein Fehler: 2 Punkte, sonst: 0 Punkte)

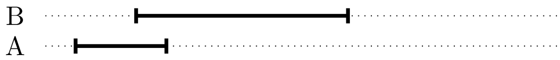
☐ 6 ☐ N(6) ☐ N(7)

c) (7 Punkte) Vervollständigen Sie die Skizze auf der vorigen Seite, sodass sie einen vollständigen Ablauf des Branch and Bound Algorithmus darstellt.

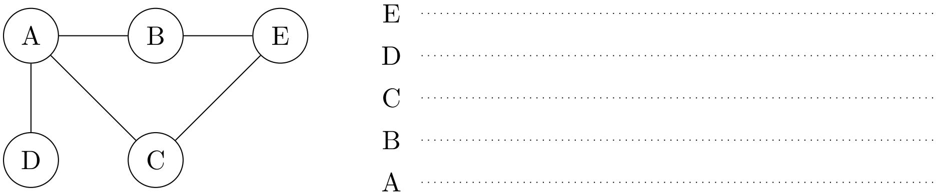
d) (7 Punkte) Zeichnen Sie den Graphen, der im grau hinterlegten Knoten betrachtet wird. Zeichnen Sie ein maximales Matching in den Graphen ein, das die untere Schranke 4 rechtfertigt. Ist es möglich ein anderes Matching zu finden, das eine bessere untere Schranke rechtfertigt?

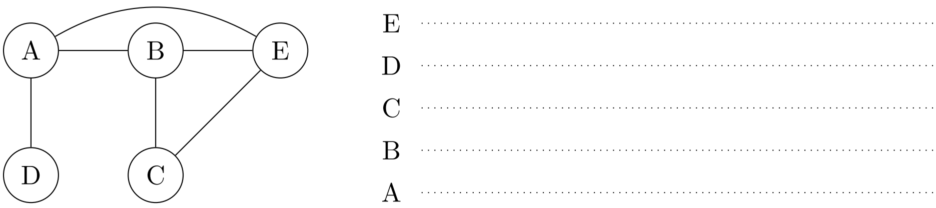
Aufgabe A3: NP-Vollständigkeit Spezialfälle (20 Punkte)

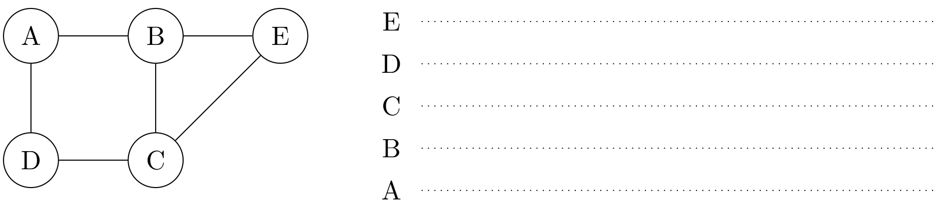
a) (9 Punkte) Gegeben sind die folgenden vier Graphen. Wenn es sich bei einem Graphen um einen **Intervallgraphen** handelt, zeichnen Sie eine passende Intervallmenge auf die vorgegebenen Linien. Zum Beispiel:



Wenn ein Graph **kein Intervallgraph** ist, entfernen Sie so wenige Knoten wie möglich, damit die Intervalleigenschaft hergestellt wird und zeichnen Sie dann vom resultierenden Graphen die Intervallmenge. Streichen Sie die entfernten Knoten im Graphen eindeutig erkennbar durch.







b) (5 Punkte) Kreuzen Sie an, ob folgende Aussagen wahr oder falsch sind.

(+1 Punkt für jede richtige, -1 Punkt für jede falsche und 0 Punkte für keine Antwort, keine negativen Punkte)

(i) Jeder Intervallgraph ist 3-färbbar.

☐ Wahr ☐ Falsch

(ii) Es gibt eine polynomielle Reduktion von 3-Color auf 5-Color.

☐ Wahr ☐ Falsch

(iii) Für Intervallgraphen liegt das Problem 3-Color in NP.

☐ Wahr ☐ Falsch

(iv) Für Intervallgraphen ist das Problem 2-Color NP-vollständig.

☐ Wahr ☐ Falsch

(v) Jeder Baum ist 3-färbbar.

☐ Wahr ☐ Falsch

c) (6 Punkte) Gegeben sind folgende Reduktionen von 2-Color nach 2-SAT. Bei jeder Reduktion wird aus einem Graphen $G = (V, E)$ eine 2-SAT Formel Φ generiert. Geben Sie für jede Reduktion an, ob diese korrekt ist. Begründung ist *nicht* notwendig.

Hinweis: Damit eine Reduktion korrekt ist, muss Φ erfüllbar sein genau dann wenn G 2-färbbar ist.

(+3 Punkte für jede richtige, -3 Punkte für jede falsche und 0 Punkte für keine Antwort, keine negativen Punkte)

(i) Für jede Kante $(v, w) \in E$ wird eine Klausel $(x_v \vee x_w)$ generiert. Für jedes Knotenpaar das nicht mit einer Kante verbunden ist wird eine Klausel $(\neg x_v \vee \neg x_w)$ generiert. Alle Klauseln konjugiert ergeben dann Φ.

Formal bedeutet das

$$\Phi = \bigwedge_{(v,w)\in E} (x_v \vee x_w) \wedge \bigwedge_{(v,w)\notin E} (\neg x_v \vee \neg x_w)$$

☐ Korrekt ☐ Inkorrekt

(ii) Für jede Kante $(v, w) \in E$ werden zwei Klauseln generiert: $(\neg x_v \vee x_w)$ und $(x_v \vee \neg x_w)$. Alle Klauseln konjugiert ergeben dann Φ.

Formal bedeutet das

$$\Phi = \bigwedge_{(v,w)\in E} \big((\neg x_v \vee x_w) \wedge (x_v \vee \neg x_w)\big)$$

☐ Korrekt ☐ Inkorrekt

Aufgabe A4: Dynamisches Programmieren **(20 Punkte)**

a) (10 Punkte) Betrachten Sie das aus der Vorlesung bekannte Rucksackproblem. Gegeben sei eine Instanz mit einer Kapazität $G = 5$ und den folgenden fünf Gegenständen, die jeweils nur einmal vorhanden sind:

	Gegenstand				
	o_1	o_2	o_3	o_4	o_5
Wert w_i	7	4	4	5	6
Gewicht g_i	4	2	1	2	3

Vervollständigen Sie die untenstehende Tabelle M, sodass der Eintrag $M[i, g]$ in Zeile i und Spalte g gerade dem Wert $OPT(i, g)$ entspricht, der sich mit den ersten i Gegenständen $o_1, \ldots, o_i$ und einem Rucksack der Kapazität g erreichen lässt.

Kapazität

Gegenstände	0	1	2	3	4	5
$\emptyset$	0	0	0	0	0	0
$\{o_1\}$	0					
$\{o_1, o_2\}$	0					
$\{o_1, o_2, o_3\}$	0					
$\{o_1, o_2, o_3, o_4\}$	0					
$\{o_1, o_2, o_3, o_4, o_5\}$	0					

(i) Was ist der optimale *Wert* einer Lösung für einen Rucksack mit Kapazität $G = 4$? Wird der Gegenstand o_5 für diese optimale Lösung benötigt?

Wert:

Wird o_5 benötigt (ja/nein)?

(ii) Wie errechnet sich der Eintrag $M[i, g]$ unter der Annahme, dass $g_i \leq g$?

$M[i, g] = \max\{$ $\}$

b) (8 Punkte) Sei nun eine weitere Instanz des Rucksackproblems mit Kapazität $G = 7$ und den folgenden sechs Gegenständen gegeben.

	Gegenstand					
	o_1	o_2	o_3	o_4	o_5	o_6
Wert	1	4	7	3	3	8
Gewicht	2	1	4	1	2	3

Die Wertetabelle M nach Ausführung des Dynamischen Programms lautet wie folgt:

Gegenstände \ Kapazität	0	1	2	3	4	5	6	7
$\emptyset$	0	0	0	0	0	0	0	0
$\{o_1\}$	0	0	1	1	1	1	1	1
$\{o_1, o_2\}$	0	4	4	5	5	5	5	5
$\{o_1, o_2, o_3\}$	0	4	4	5	7	11	11	12
$\{o_1, o_2, o_3, o_4\}$	0	4	7	7	8	11	14	14
$\{o_1, o_2, o_3, o_4, o_5\}$	0	4	7	7	10	11	14	14
$\{o_1, o_2, o_3, o_4, o_5, o_6\}$	0	4	7	8	12	15	15	18

(i) Kreisen Sie in der Tabelle all jene Felder ein, die der Algorithmus `Find-Solution(M)` aus der Vorlesung bei der Berechnung der Lösungsmenge S ausliest und verwendet.

(ii) Geben Sie die Menge der Gegenstände in der Lösungsmenge S an.

$S = \{$ ________________ $\}$

c) (2 Punkte) Sei eine Instanz des Rucksackproblems mit n Gegenständen und Kapazität G des Rucksacks gegeben. Kreuzen Sie *alle* gültigen Laufzeitschranken für den Algorithmus `Find-Solution(M)`, der die ausgefüllte Wertetabelle M als Eingabe erhält.

(2 Punkte, wenn alle Kreuze korrekt sind, 1 Punkt bei genau einem Fehler, ansonsten 0 Punkte)

- ☐ $O(n)$
- ☐ $O(G)$
- ☐ $O(\min\{n, G\})$
- ☐ $O(n \cdot G)$
- ☐ $O(n + G)$

Aufgabe A5: Approximationsalgorithmen **(20 Punkte)**

a) (15 Punkte) Für einen gerichteten Graphen $G = (V, E)$ wird eine größtmögliche (bezüglich Kardinalität) Teilmenge $F \subseteq E$ von Kanten gesucht, die es erlaubt eine topologische Sortierung für den reduzierten Graphen $G' = (V, F)$ zu finden.

(i) Geben Sie für dieses Problem einen polynomiellen Approximationsalgorithmus mit Gütegarantie $1/2$ an. Eine eindeutige Beschreibung in wenigen Worten reicht, Pseudocode ist nicht notwendig.

Hinweis: Betrachten Sie eine beliebige lineare Ordnung der Knotenmenge. Welche zwei Arten von Kanten können Sie unterscheiden?

(ii) Argumentieren Sie, warum Ihr Algorithmus immer eine korrekte Lösung liefert und die geforderte Gütegarantie einhält.

b) (5 Punkte) Kreuzen Sie die richtigen Aussagen an:
(+1 Punkt für jede richtige, -1 Punkt für jede falsche und 0 Punkte für keine Antwort, keine negativen Punkte)

- Sei v^* der Wert einer optimalen Lösung eines Minimierungsproblems. Ein Approximationsalgorithmus mit Gütegarantie α liefert immer eine Lösung mit einem Wert $\leq v^* + \alpha$.

  ☐ Wahr ☐ Falsch

- Ein Approximationsalgorithmus mit Gütegarantie 2 kann eine Lösung zurückliefern, deren Wert 2% über dem optimalen Lösungswert liegt.

  ☐ Wahr ☐ Falsch

- Ein Approximationsalgorithmus mit Gütegarantie 1 liefert immer eine optimale Lösung.

  ☐ Wahr ☐ Falsch

- Sei v^* der Wert einer optimalen Lösung eines Maximierungsproblems. Ein Approximationsalgorithmus mit Gütegarantie α liefert immer eine Lösung mit einem Wert $\geq v^* \cdot \alpha$.

  ☐ Wahr ☐ Falsch

- Die aus der Vorlesung bekannte Spannbaumheuristik löst das *Euklidische Traveling Salesperson Problem* mit Gütegarantie 3.

  ☐ Wahr ☐ Falsch