TU Wien
Institut für Logic and Computation
Algorithms and Complexity Group

186.866 Algorithmen und Datenstrukturen VU 8.0

2. Test, 2024S

28. Juni 2024

Gruppe A

Nachfolgende Angaben gut **leserlich in BLOCKSCHRIFT** machen.

Nachname: ____________ Vorname: ____________

Matrikelnummer: ____________ Unterschrift: ____________

Sie dürfen die Lösungen nur auf diesen Prüfungsbogen schreiben. Es ist nicht zulässig, eventuell mitgebrachtes eigenes Papier zu verwenden. Benutzen Sie dokumentenechte Stifte (keine Bleistifte, etc.).

Die Verwendung von Taschenrechnern, Mobiltelefonen, Smartwatches, 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. Schreiben Sie leserlich, 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 und NP, Spezialfälle **(______ / 20 Punkte)**

a) (8 Punkte) Seien A, B, C und D Ja/Nein-Probleme und n jeweils die Eingabegröße. Nehmen Sie an, es gibt

- eine Reduktion von 3-SAT nach A in Zeit $O(n^5)$,
- eine Reduktion von A nach B in Zeit $O(n^2)$,
- eine Reduktion von A nach 3-SAT in Zeit $O(n^3)$,
- eine Reduktion von A nach C in Zeit $O(2^n)$,
- eine Reduktion von C nach D in Zeit $O(n^2 \log(n))$,
- einen Algorithmus für D mit Laufzeit $O(n^5)$.

(i) Geben Sie die engste obere Schranke für die Laufzeit einer Reduktion von 3-SAT nach B in O-Notation an, die sich aus diesen Annahmen ableiten lässt. Begründen Sie Ihre Antwort kurz.

(ii) Was folgt aus den obigen Annahmen für die Komplexität der Probleme A, B und C? Beantworten Sie die Frage, indem Sie **alle** zutreffenden Felder in der folgenden Tabelle ankreuzen.

	in P	in NP	NP-schwer	NP-vollständig
A				
B				
C				

b) (8 Punkte) Wir betrachten folgendes Problem unter der Annahme P $\neq$ NP.

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

Welche der folgenden Zertifikate sind (für einen geeigneten Zertifizierer) geeignet, um zu zeigen, dass das gegebene Problem in NP ist? Nehmen Sie dabei an, der Eingabegraph G ist von der Form wie in den Spalten angegeben. Kreuzen Sie geeignete Kombinationen an und begründen Sie Zeilen mit nur einem Kreuz kurz.

	G ist schlicht	G ist ein Baum
Ein leerer String.		
Alle Independet Sets in G.		
Ein Independent Set S der Größe k.		
Ein Vertex Cover C mit $n - k$ Knoten.		

c) (4 Punkte) Betrachten Sie folgendes Problem und Zertifikat:

Problem: Gegeben sei ein Graph G mit n Knoten und m Kanten. Gibt es eine *ausgeglichene* 3-Färbung φ von G, das heißt, eine Färbung der Knoten von G in Rot, Gelb und Blau, sodass benachbarte Knoten nicht die gleiche Farbe besitzen und die Anzahl der rot, gelb und blau gefärbten Knoten sich jeweils um maximal eins unterscheidet?

Zertifikat: Eine ausgeglichene 3-Färbung φ von G.

(i) Geben Sie für das gegebene Zertifikat einen geeigneten Zertifizierer an, um zu zeigen, dass das Problem in NP ist.

(ii) Erklären Sie, welche Eigenschaften ein Zertifizierer im Allgemeinen erfüllen muss, um für ein geeignetes Zertifikat zu zeigen, dass ein Problem in NP ist.

Aufgabe A2: Branch-and-Bound (______ / 20 Punkte)

Für die folgenden Teilaufgaben betrachten wir das Talent Scheduling Minimierungsproblem. Dabei soll die Reihenfolge, in der Szenen für einen Film gedreht werden, bestimmt werden, sodass die Kosten minimal sind. Pro Tag kann nur eine Szene gedreht werden und alle Talente ($a_i \in A$) sind von Drehbeginn des Films durchgehend bis zum Dreh ihrer letzten Szene ($s_j \in S$) angestellt. Das Tagesgehalt ($l_i \in L$) der Talente sowie die Beteiligung an einzelnen Szenen kann tabellarisch wie in der unten stehenden Tabelle dargestellt werden.

Beispielsweise verursacht die Szenenreihenfolge [1,2,3,4] Kosten von 850, da erst nach Drehtag 3, Talente A und D mit ihrem Dreh fertig sind und für den letzten Drehtag noch Talente B und C bezahlt werden müssen.

Szene Nr. / Talent	1	2	3	4	Gehalt
A	1	1	1	0	100
B	0	0	1	1	50
C	1	0	1	1	50
D	0	0	1	0	50

a) (4 Punkte) Bestimmen Sie eine Lösung mit einem Lösungswert von 750 oder weniger für die oben gegebene Instanz. Es muss kein Lösungsweg angegeben werden.

b) (4 Punkte) Ein Branch-and-Bound Algorithmus soll das Talent Scheduling Problem lösen. Ein Teilproblem wird dabei durch ein Tupel (T, F) repräsentiert, in dem T die gegebene Probleminstanz ist und F die bisher fixierte Szenenreihenfolge. Beschreiben Sie, wie der Algorithmus das Teilproblem (T, F) in kleinere Teilprobleme zerteilen kann. Eine kurze und klare Beschreibung der Idee ohne Begründung genügt.

Hinweis: Beim Branching können auch mehr als 2 Kindknoten pro Elternknoten erzeugt werden.

c) (8 Punkte) Sei F eine Teillösung aus Unteraufgabe b). Kreuzen Sie an, ob die beschriebene Methode eine untere Schranke, obere Schranke oder keine Schranke für das Talent Scheduling Problem ist.
(+1 Punkt für jede richtige, -1 Punkt für jede falsche und 0 Punkte für keine Antwort, keine negativen Gesamtpunkte auf diese Unteraufgabe)

	untere	obere	keine
Die Gesamtkosten sind die Kosten für F, plus der Summe aller übrigen Szenenkosten. Die Kosten einer Szene sind die Summe der Gehälter aller beteiligten Talente.	☐	☐	☐
Fixiere eine zufällige Reihenfolge für alle noch nicht fixierten Szenen. Berechne daraus mit F die Gesamtkosten.	☐	☐	☐
Das bisher zu zahlende Gehalt für die bereits fixierten Szenen F.	☐	☐	☐
Eine Greedy Methode, die aus einer Teillösung F eine gültige Lösung erzeugt und aus dieser die Gesamtkosten bestimmt.	☐	☐	☐
Bestimme die minimale Anzahl an übrigen Drehtagen für jedes Talent. Berechne daraus mithilfe von F die Gesamtkosten.	☐	☐	☐
Zuzüglich den Kosten von F, bezahle alle Talente, die den Dreh noch nicht abgeschlossen haben bis ans Ende des Films.	☐	☐	☐
Die Kosten für F plus, Restkosten als die Anzahl der noch nicht gedrehten Szenen, mal dem Durchschnittsgehalt aller Talente.	☐	☐	☐
Berechne die Gesamtkosten als die Anzahl der bereits gedrehten Szenen multipliziert mit der Summe aller Gehälter.	☐	☐	☐

d) (4 Punkte) Kreuzen Sie an, ob die folgenden Aussagen wahr oder falsch sind.

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

(Q1) Das Branching einer Teilinstanz in einem Maximierungsproblem wird abgebrochen, wenn die lokale obere Schranke kleiner ist als die globale untere Schranke.

☐ Wahr ☐ Falsch

(Q2) Eine korrekte Lösung kann nur gefunden werden, wenn alle Teilprobleme betrachtet werden.

☐ Wahr ☐ Falsch

(Q3) Branch-and-Bound Methoden bringen immer eine Laufzeitverbesserung gegenüber Brute-Force Methoden.

☐ Wahr ☐ Falsch

(Q4) Bei der Best-first Durchmusterung in einem Maximierungsproblem wird als nächstes das Teilproblem bearbeitet, welches die geringste Differenz zwischen lokaler unterer und oberer Schranke hat.

☐ Wahr ☐ Falsch

Aufgabe A3: Dynamisches Programmieren **(______ / 20 Punkte)**

a) (5 Punkte) Wir betrachten eine Funktion $f\colon \mathbb{N} \to \mathbb{Z}$, die wie folgt definiert ist:

$$f(n) = \begin{cases} 1 & \text{wenn } n = 1 \text{ oder } n = 2, \\ 3 \cdot f(n-1) - f(n-2) & \text{wenn } n > 2 \text{ und } n \text{ gerade}, \\ f(n-1) + f(n-2) & \text{wenn } n > 2 \text{ und } n \text{ ungerade}. \end{cases}$$

Bestimmen Sie die Werte $f(5)$, $f(7)$ und $f(11)$.

Hinweis: Nutzen Sie einen geeigneten Algorithmus. Notieren Sie Ihre Zwischenergebnisse.

b) (8 Punkte) Das Problem Partition ist wie folgt definiert:

Partition

Eingabe: Eine Menge $A = \{a_1, a_2, \ldots, a_n\}$ mit n positiven, ganzen Zahlen, die sich auf eine gerade Zahl S aufsummieren, also $S = \sum_{i=1}^{n} a_i$.

Aufgabe: Entscheiden Sie, ob sich A in zwei gleich große Teile aufteilen lässt, das heißt, ob eine Teilmenge $B \subseteq A$ existiert, sodass $S/2 = \sum_{b \in B} b$.

Geben Sie einen Algorithmus an, der Partition löst und auf dynamischer Programmierung beruht. Seine Laufzeit muss in $O(nS)$ liegen. Die Ausgabe soll 1 sein, wenn eine Lösung existiert und 0 sonst. Vervollständigen Sie dafür den nachfolgenden Pseudocode auf der nächsten Seite.

Hinweis: Sie können das aus der Vorlesung bekannte dynamische Programm für das Rucksackproblem anpassen.

```
SolvePartition(Menge A):
S ← ∑_{i=1}^{n} a_n
M[0,0] ← 1
for s ← 1 bis S/2
    M[0,s] ← 0

for i ← 1 bis n
```


```
return M[n, S/2]
```

c) (7 Punkte) Gegeben ist ein gerichteter Graph $G = (V, E)$ mit $|V| = n$ Knoten. Die Kanten von G sind mit Kantengewichten $c_{vw} \in \mathbb{R}$ für alle Kanten $(v, w) \in E$ gewichtet.

(i) Gehen Sie davon aus, dass G keine negativen Kreise enthält. Ergänzen Sie Bellmans Rekursionsgleichungen für die Länge $OPT(i, v)$ eines kürzesten v-t Pfades in G mit höchstens i Kanten, wobei $v \in V$ ein beliebiger Knoten ist und $t \in V$ ein fester Zielknoten:

$OPT(0, t) = 0,$

$OPT(0, v) = \infty$ für $v \neq t$, und

$OPT(i, v) = \boxed{}$ für $i \geq 1$.

(ii) Geben Sie den kleinsten Wert für die Kantenanzahl i an, sodass für jeden Graphen G, der keinen negativen Kreis enthält, Folgendes gilt:

$$OPT(i, v) = OPT(i + 1, v) \text{ für alle Knoten } v \in V.$$

Der kleinste Wert ist $i = \boxed{}$.

(iii) Erklären Sie *kurz*, wie ein Algorithmus mithilfe der obigen Gleichungen erkennen kann, ob Graph G einen negativen Kreis enthält.

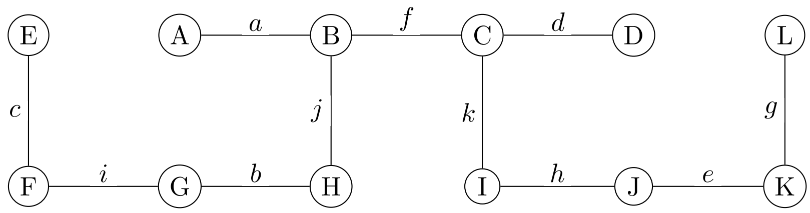
Aufgabe A4: Approximationsalgorithmen **(______ / 20 Punkte)**

a) (4 Punkte) Sei A ein ε-Approximationsalgorithmus für ein Minimierungsproblem, der für jede Instanz x eine gültige Lösung mit Lösungswert $c_A(x) > 0$ liefert. Sei $c_{opt}(x) > 0$ der Wert einer optimalen Lösung.

- Geben Sie die Ungleichung für die Gütegarantie an.

- Welche Werte kann ε bei Minimierungsproblemen annehmen?

b) (8 Punkte) Beim MINIMUM VERTEX COVER Problem geht es darum, für einen gegebenen Graphen ein Vertex Cover mit minimaler Anzahl an Knoten zu finden. Betrachten Sie dazu den aus der Vorlesung bekannten 2-Approximationsalgorithmus und den folgenden Graphen:



(i) (4 Punkte) Führen Sie den Algorithmus auf dem gegebenen Graphen aus und geben Sie das resultierende Vertex Cover an. Wenn Sie die Auswahl zwischen mehreren Kanten haben, nehmen Sie die alphabetisch kleinste.

(ii) (2 Punkte) Geben Sie ein optimales Vertex Cover an.

(iii) (2 Punkte) Welche Kanten muss der Approximationsalgorithmus auswählen, um ein optimales Vertex Cover zu finden?.

c) (4 Punkte) Wir betrachen das aus der Vorlesung bekannte MAX-SAT Problem (maximiere die Anzahl der erfüllten Klauseln in einer booleschen Formel) und einen Approximationsalgorithmen A für MAX-SAT mit einer Gütegarantie von $\frac{3}{4}$.

(i) Gegeben ist eine Formel F mit 40 Klauseln, sodass die optimale Variablenzuweisung 32 Klauseln in F erfüllt. Weiters ist I eine Variablenzuweisung die mittels A für F gefunden wurde. In welchem Intervall befindet sich die Anzahl der von I erfüllten Klauseln? Geben Sie die engstmöglichen Schranken an.

(ii) Angenommen A findet für eine andere Formel F' mit 50 Klauseln eine Variablenzuweisung die 33 Klauseln in F' erfüllt. In welchem Intervall befindet sich eine optimale Lösung für F'? Geben Sie die engstmöglichen Schranken an.

d) (4 Punkte) Kreuzen Sie an, ob die folgenden Aussagen wahr oder falsch sind.

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

(Q1) Ein Approximationsalgorithmus mit Gütegarantie 4 kann eine Lösung zurückliefern, deren Wert 4% über dem optimalen Lösungswert liegt.

☐ Wahr ☐ Falsch

(Q2) Die aus der Vorlesung bekannte Spannbaumheuristik löst das *symmetrische Traveling Salesperson Problem* mit Gütegarantie 3.

☐ Wahr ☐ Falsch

(Q3) Falls P = NP gilt, dann existiert für das LASTVERTEILUNG Problem ein polynomieller 1.5-Approximationsalgorithmus.

☐ Wahr ☐ Falsch

(Q4) Ein $\frac{1}{2}$-Approximationsalgorithmus für ein Maximierungsproblem findet eine Lösung die mindestens halb so groß wie die optimale Lösung ist.

☐ Wahr ☐ Falsch

Aufgabe A5: Heuristische Verfahren **(______ / 20 Punkte)**

a) (8 Punkte) Wir betrachten eine lokale Suche für das aus der Vorlesung bekannte MAX-SAT Problem (maximiere die Anzahl der erfüllten Klauseln in einer Boole'schen Formel). Variablenzuweisungen werden als binäre Vektoren $(x_1, \ldots, x_n)$ repräsentiert. N bezeichnet die aus der Vorlesung bekannte 1-flip Nachbarschaft für binäre Vektoren.

Gegeben ist eine Boole'sche Formel F mit Variablen x_1, x_2, x_3:

$$(x_1 \vee x_2) \wedge (\overline{x_1} \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_2 \vee \overline{x_3})$$

Weiters ist die Ausgangslösung $I = (1, 0, 1)$ gegeben, es wird also x_1 auf *wahr*, x_2 auf *falsch* und x_3 auf *wahr* gesetzt.

(i) Bestimmen Sie die Nachbarschaft $N(I)$ von I und geben Sie für jeden Vektor $J \in N(I)$ an, wie viele Klauseln in F von J erfüllt werden.

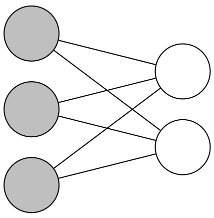
(ii) Welcher Vektor in $N(I)$ wird bei einer lokalen Suche als nächstes ausgewählt, wenn Best Improvement als Schrittfunktion verwendet wird?

(iii) Angenommen die Lösungen in $N(I)$ werden in lexikographisch aufsteigender Reihenfolge durchmustert. Es würde also $(0, 0, 0)$ vor $(0, 0, 1)$ vor $(0, 1, 0)$ vor $(0, 1, 1)$ usw. durchmustert werden. Welcher Vektor in $N(I)$ wird dann bei einer lokalen Suche als nächstes ausgewählt, wenn First Improvement als Schrittfunktion verwendet wird?

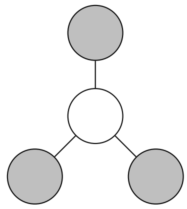
b) (8 Punkte) Im Folgenden bezeichnen N und N' die zwei aus der Vorlesung bekannten Nachbarschaftstrukturen für das Minimum Vertex Cover Problem:

- $S \in N(C)$ wenn S ein Vertex Cover ist das durch Löschen eines einzigen Knotens aus C erzeugt werden kann.
- $S \in N'(C)$ wenn $S \in N(C)$ oder wenn S ein Vertex Cover ist das durch Hinzufügen eines Knotens von $V \setminus C$ und Löschen von zwei Knoten aus C gebildet werden kann.

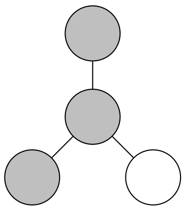
Mit welchen Nachbarschaftsstrukturen kann eine lokale Suche für Minimum Vertex Cover bei den folgenden Graphen und Ausgangslösungen (grau markiert) ein globales Optimum finden? Kreuzen Sie alle zutreffenden Felder an.



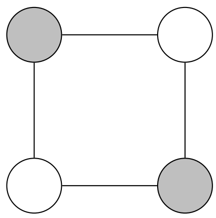
(i) ☐ N ☐ N' ☐ weder noch



(ii) ☐ N ☐ N' ☐ weder noch



(iii) ☐ N ☐ N' ☐ weder noch



(iv) ☐ N ☐ N' ☐ weder noch

c) (4 Punkte) Kreuzen Sie an, ob die folgenden Aussagen wahr oder falsch sind.

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

(Q1) Falls P = NP gilt, liefert eine lokale Suche für Max-Sat mit 1-flip Nachbarschaft garantiert eine optimale Lösung und zwar in Polynomialzeit und unabhängig von der Ausgangslösung.

☐ Wahr ☐ Falsch

(Q2) Bei der Tabu-Suche wird das Metropolis-Kriterium verwendet, um das erneute Besuchen bereits gefundener Lösungen zu vermeiden.

☐ Wahr ☐ Falsch

(Q3) Im Allgemeinen gilt: wenn man lokale Suche mit Schrittfunktion First Improvement lange genug ausführt, findet man immer ein lokales Optimum.

☐ Wahr ☐ Falsch

(Q4) Bei Simulated Annealing werden immer nur Nachbarlösungen ausgewählt, welche strikt besser sind als die momentan ausgewählte Lösung.

☐ Wahr ☐ Falsch