



CHRISTIAN-ALBRECHTS-UNIVERSITÄT ZU KIEL

Institut für Informatik, Arbeitsgruppe Algorithmen und Komplexität
Prof. Dr. K. Jansen

22. Juli 2025

Modulprüfung zur Vorlesung »Analyse von Algorithmen und Komplexität – Komplexitätsteil«

SS 2025

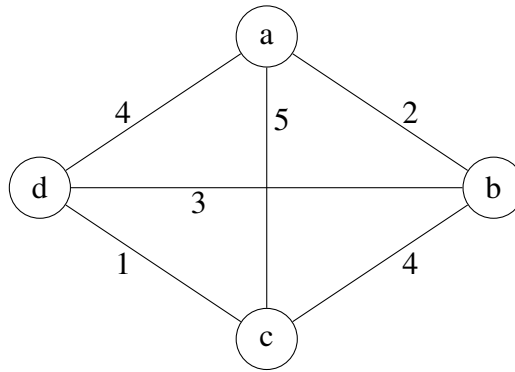
Name: _____ Matrikel-Nr.: _____

Hinweise:

- Bearbeiten Sie *alle* Aufgaben.
- Sie haben 10 Minuten Einlesezeit. Danach haben Sie **120 Minuten** Zeit für die Klausurbearbeitung.
- Erlaubte Hilfsmittel: Ein doppelseitig handschriftlich beschriebenes Blatt Papier DIN A4, (bunte) Stifte (kein Rot, kein Grün, kein Bleistift).
- Einsichtnahme: **2. September 2025, 10 bis 11 Uhr**. Der Raum für die Einsicht wird noch bekannt gegeben.
- Die Klausur besteht aus **5 Aufgaben**, die alle **10 Punkte** wert sind.
- Zum Bestehen müssen mindestens **18 Punkte** erreicht werden.
- Aufgabe 5 gibt zusätzlich **10 Bonuspunkte**. Insgesamt können also **50 Punkte plus 10 Bonuspunkte** erreicht werden.

Aufgabe 1 ANWENDUNG – CHRISTOFIDES' ALGORITHMUS (6 + 4 Punkte)

Gegeben sei folgender Graph $G = (V, E)$:



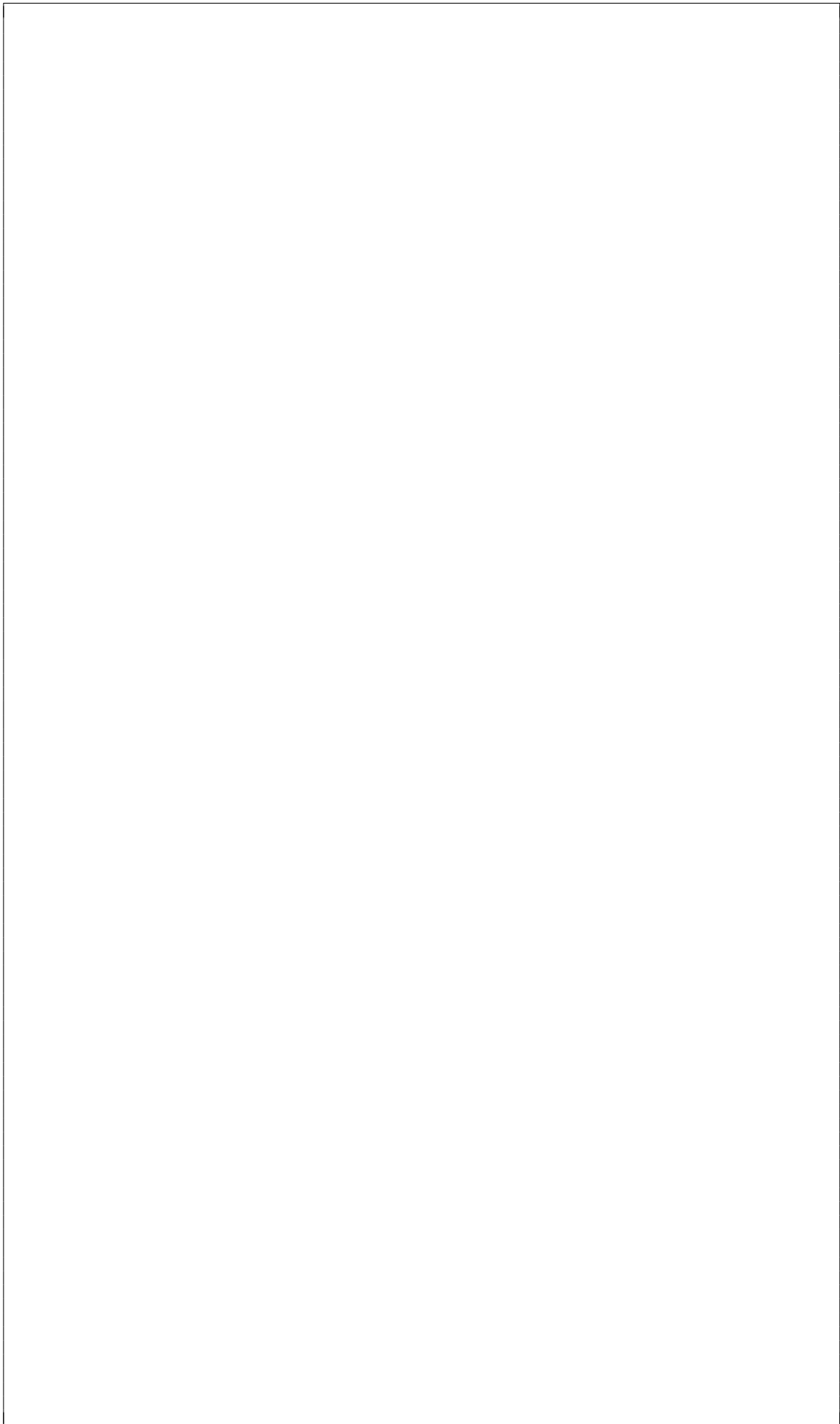
- a) Wenden Sie den Algorithmus von Christofides (ΔTSP_2) auf den Graphen G mit Startknoten a an. Geben Sie dabei alle Graphen an, die in Zwischenschritten entstehen. Erwähnen Sie ebenfalls Schritte, die bei dieser Anwendung zu keiner Änderung führen, die der Algorithmus aber überprüfen muss.



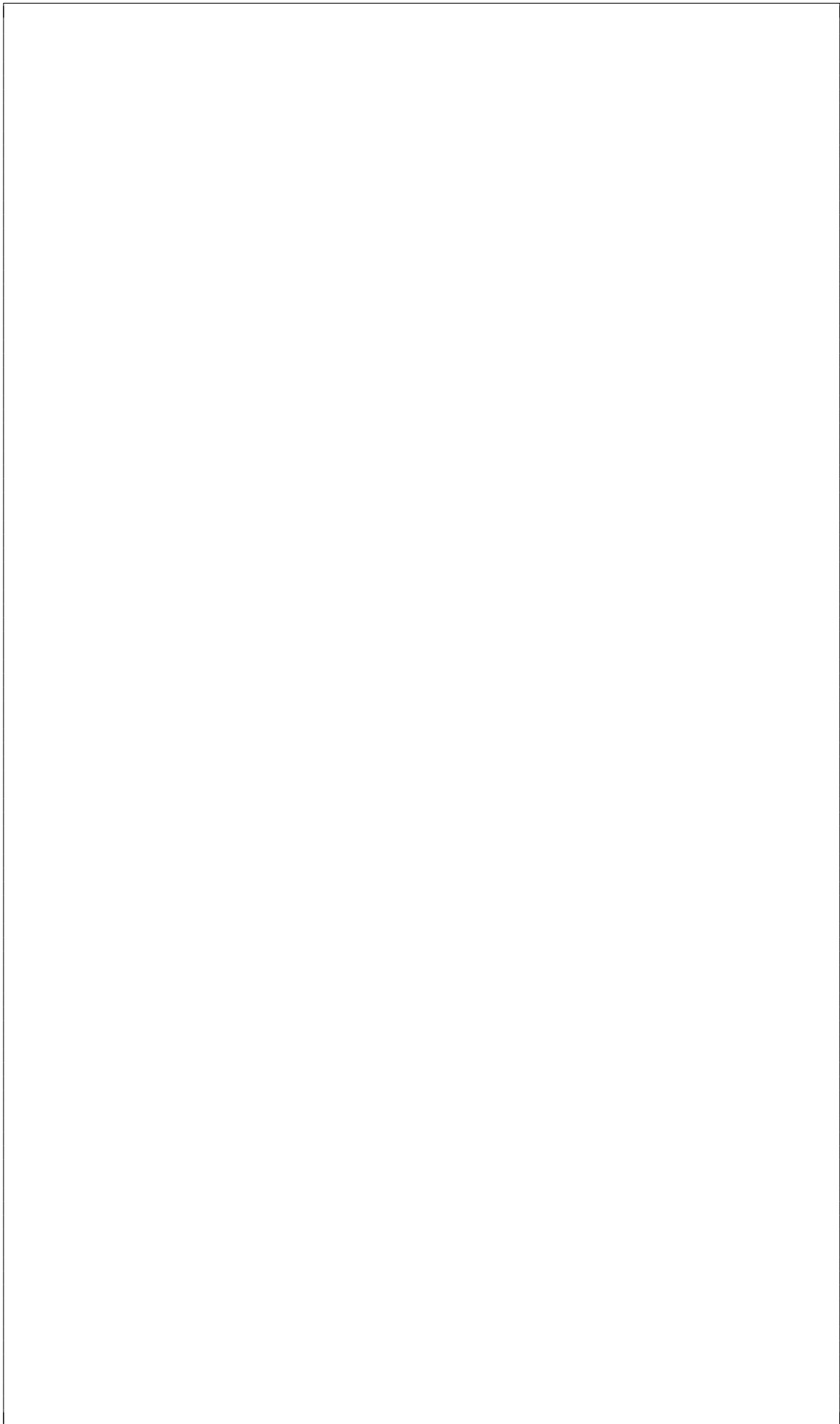
Matrikel-Nr.:

Name:

Name: _____ Matrikel-Nr.: _____



Name: _____ Matrikel-Nr.: _____



- b) Für die Korrektheit des Algorithmus' von Christofides werden zwei zusätzliche Eigenschaften für die Eingabe des Travelling Salesperson Problems benötigt. Nennen Sie die Eigenschaften und geben Sie jeweils eine Eingabe an, welche die Eigenschaft *nicht* erfüllt. Sie dürfen auch eine Eingabe nennen, welche beide Eigenschaften nicht erfüllt. Geben Sie auch die Güte des Algorithmus' an.



Aufgabe 2 VORLESUNGSBEWEIS – DEFINITION NP (2 + 8 Punkte)

Beweisen Sie folgende Aussage:

$$\{L \subseteq \Sigma^* \mid \exists \text{ NDTM } M \text{ über Alphabet } \Gamma \supset \Sigma \text{ und ein Polynom } T \text{ mit } M \text{ akzeptiert } L \\ \text{ in nicht-deterministischer Zeit } T\} \subseteq \{L \subseteq \Sigma^* \mid \exists \text{ polynomieller Verifizierer für } L\}$$

Ausformuliert bedeutet das: Sei $L \subseteq \Sigma^*$ also eine Sprache, für die eine NDTM M über einem Alphabet $\Gamma \supset \Sigma$ und ein Polynom T existiert, sodass L von M in nicht-deterministischer Zeit T entschieden wird.

Nun müssen Sie folgende Aussage beweisen: *Es existiert ein polynomieller Verifizierer für L .*

- a) Geben Sie dafür zunächst für die Sprache L , die durch folgende NDTM M akzeptiert wird, einen polynomiellen Verifizierer an, der in analoger Weise arbeitet. Die NDTM M erhält als Eingabe einen ungerichteten Graphen $G = (V, E)$ und eine Zahl $k \in \mathbb{N}_0$.

$M(G = (V, E), k)$

1. Rate nicht-deterministisch eine Menge von Knoten $C \subseteq V$.
2. Für alle $\{u, v\} \in E$: Falls $u \notin C$ und $v \notin C$, verwerfe.
3. Falls $|C| > k$, verwerfe.
4. Akzeptiere.



Matrikel-Nr.:

Name:

b) Führen Sie nun einen allgemeingültigen Beweis.

Name: _____ Matrikel-Nr.: _____



Aufgabe 3 NP-VOLLSTÄNDIGKEIT (2 + 8 Punkte)

Wir betrachten folgendes Problem:

Problem: k -CLIQUE-DEG-3

Eingabe: Ein ungerichteter Graph $G = (V, E)$, sodass jeder Knoten mindestens Grad 3 hat, also $\deg(v) \geq 3$ für alle $v \in V$. Weiter ist eine Zahl $k \in \mathbb{N}_{>0}$ gegeben.

Entscheide: Existiert eine Clique $C \subseteq V$ mit $|C| \geq k$.

- a) Zeigen Sie, dass k -CLIQUE für $k \in \{1, 2, 3, 4\}$ in polynomieller Zeit lösbar ist durch Angabe von polynomiellen Algorithmen.

Matrikel-Nr.:

Name:

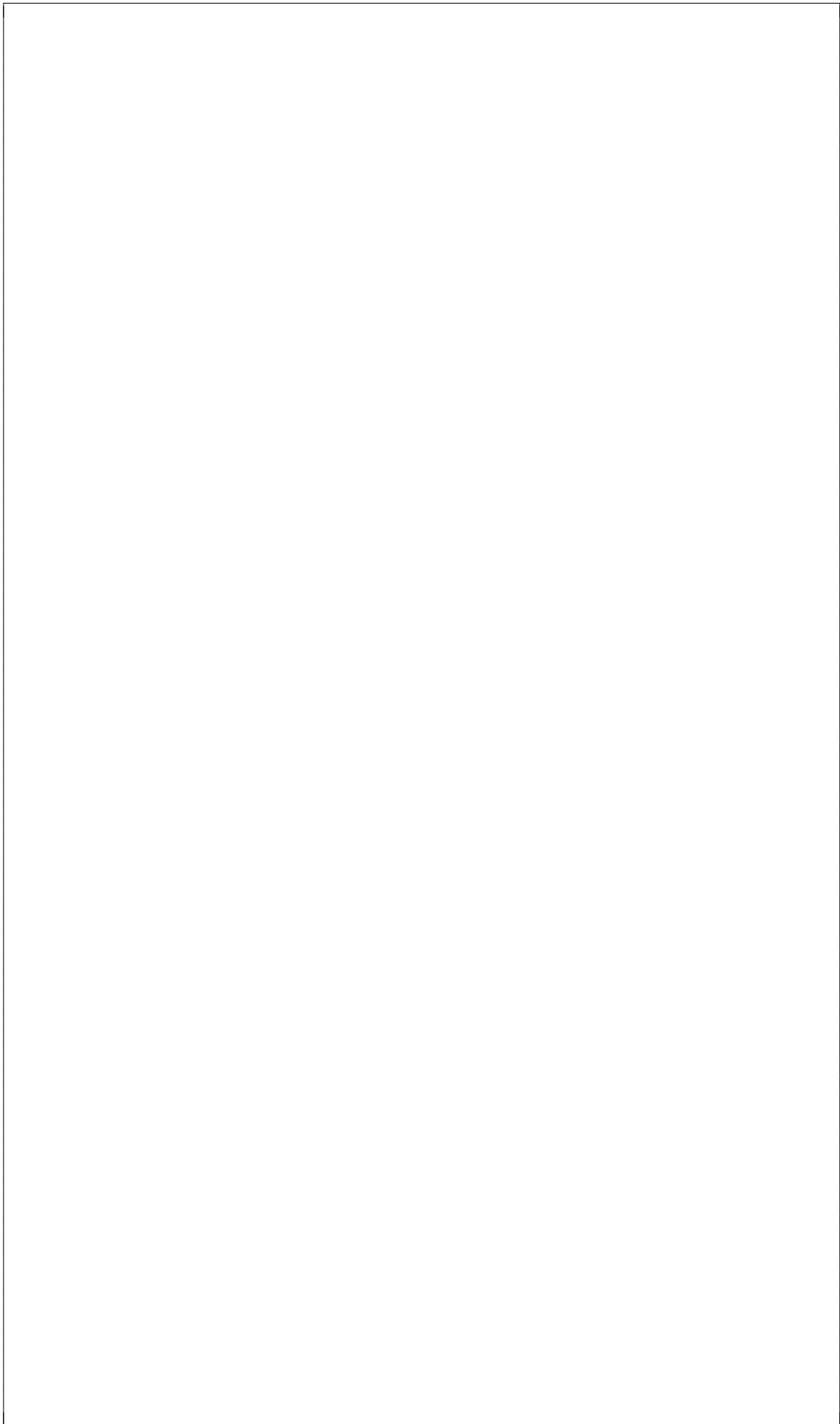


b) Zeigen Sie, dass k -CLIQUE-DEG-3 NP-vollständig ist.

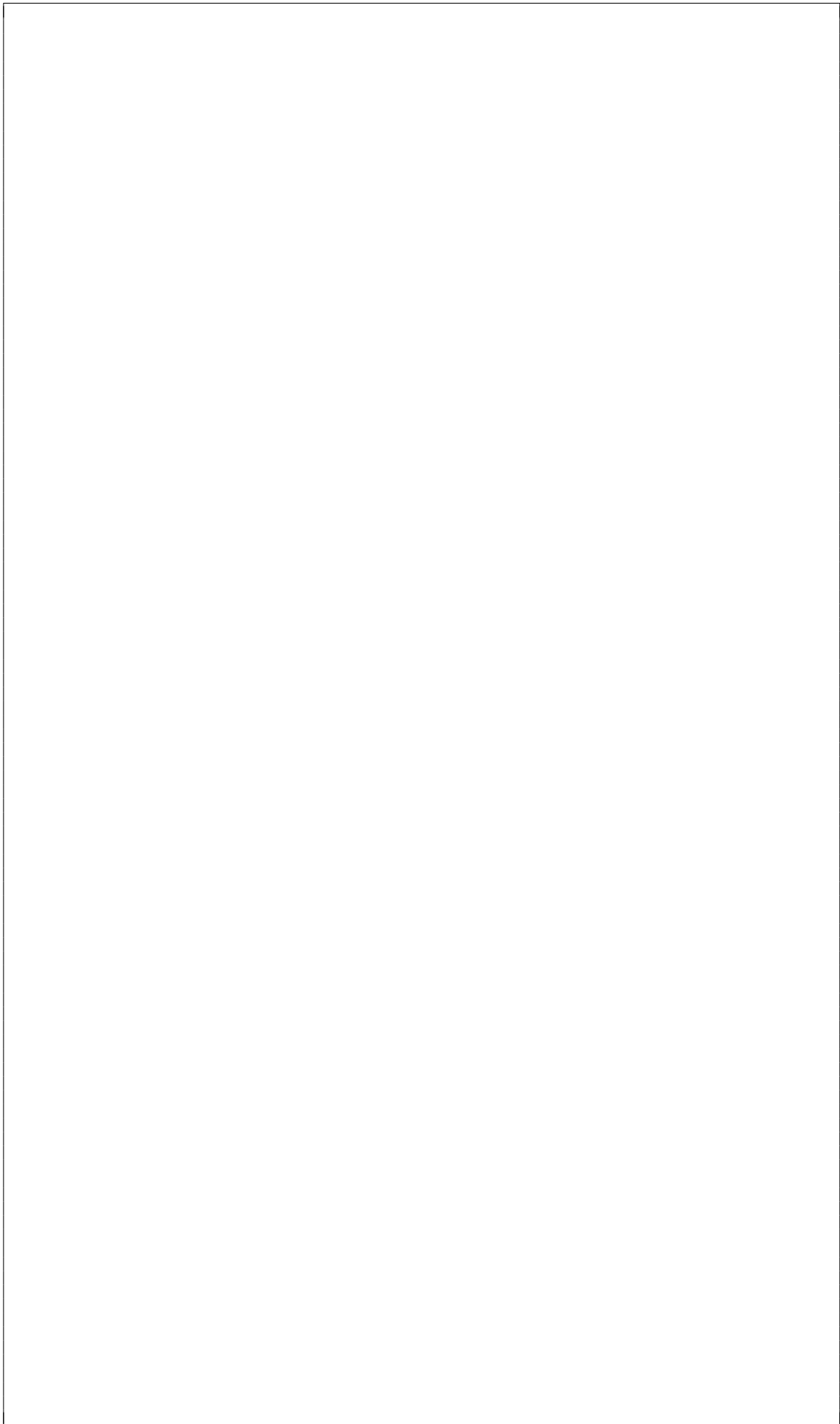
Name: _____ Matrikel-Nr.: _____



Name: _____ Matrikel-Nr.: _____



Name: _____ Matrikel-Nr.: _____



Aufgabe 4 ETH (1+4+1+4 Punkte)

Problem: NAE-4-SAT

Eingabe: n Variablen $x_1, \dots, x_n$ und m Klauseln $C_1, \dots, C_m$, wobei $|C_i| \leq 4$ für alle $i \in [m]$.

Entscheide: Existiert eine Belegung φ , sodass für jede Klausel C_i Literale $y, z \in C_i$ mit $\varphi(y) \neq \varphi(z)$ existieren.

Betrachten Sie folgende Reduktion von 3-SAT auf NAE-4-SAT. Für eine gegebene 3-SAT Instanz mit Variablen $x_1, \dots, x_n$ und Klauseln $C_1, \dots, C_m$ konstruieren wir eine NAE-4-SAT Instanz wie folgt. Die Variablen sind $x_1, \dots, x_n, w$, wobei w eine frische Variable ist, also $w \neq x_i$ für alle $i \in [n]$. Für jedes $i \in [m]$ nehmen wir die Klausel $C_i \cup \{w\}$ hinzu.

- Geben Sie die Anzahl der Variablen und die Anzahl der Klauseln für die resultierende NAE-4-SAT Instanz in Abhängigkeit von der originalen 3-SAT Instanz an.
- Beweisen Sie Lower Bounds für NAE-4-SAT bezüglich Anzahl der Variablen und Anzahl der Klauseln basierend auf der ETH und der oben beschriebenen Reduktion.

Matrikel-Nr.:

Name:



Problem: SETSPLITTING**Eingabe:** Eine Grundmenge S und Teilmengen $F_1, \dots, F_r \subseteq S$.**Entscheide:** Existiert eine Partition von S in S_1 und S_2 ($S = S_1 \dot{\cup} S_2$), sodass $F_i \cap S_1 \neq \emptyset \neq F_i \cap S_2$ für alle $i \in [r]$.

Betrachten Sie folgende Reduktion von NAE-4-SAT auf SETSPLITTING. Für eine NAE-4-SAT Instanz mit n Variablen $x_1, \dots, x_n$ und m Klauseln $C_1, \dots, C_m$ definiere $S = \{x_i, \bar{x}_i \mid i \in [n]\}$ als Grundmenge. Die Teilmengen $F_1, \dots, F_r$ ergeben sich folgendermaßen. Für jedes $i \in [n]$ nehmen wir die Teilmenge $\{x_i, \bar{x}_i\} \subseteq S$. Zusätzlich fügen wir für jede Klausel C_i aus NAE-4-SAT die Teilmenge $C_i \subseteq S$ hinzu.

- c) Geben Sie die Größe der Grundmenge S und die Anzahl der Teilmengen r für die resultierende SETSPLITTING Instanz in Abhängigkeit von der originalen NAE-4-SAT Instanz an.
- d) Beweisen Sie Lower Bounds für SETSPLITTING bezüglich Größe der Grundmenge und Anzahl der Teilmengen basierend auf der ETH und der oben beschriebenen Reduktion.



Aufgabe 5 APPROXIMATIVE ALGORITHMEN – SCHEDULING (1 + 1 + 8 + (10) Punkte)

Wir betrachten das *Parallel Task Scheduling* Problem. Dabei sind n Jobs J und m Maschinen gegeben. Hier hat jeder Job $j \in [n]$ eine Ausführungszeit $p_j \in \mathbb{Q}_{>0}$ und belegt dabei gleichzeitig $q_j \in [m]$ Maschinen. Es wird ein valider Schedule (keine zwei verschiedenen Jobs belegen gleichzeitig dieselbe Maschine) gesucht mit minimaler Makespan, also ein Schedule, bei dem alle Jobs möglichst früh abgearbeitet sind.

Algorithmus LISTSCHEDULING(J, m)

```

1  foreach  $i \in [m]$  do
2     $C_i \leftarrow 0$ 
3  od
4  foreach  $j \in [n]$  do
5    Sortiere Maschinen, so dass  $C_i \leq C_{i+1}$ , für alle  $i \in [m-1]$ 
6    Platziere Job  $j$  zum frühestmöglichen Zeitpunkt  $C' \leftarrow \max_{i \leq q_j} \{C_i\}$ 
7    foreach  $i \in [q_j]$  do
8       $C_i \leftarrow C' + p_j$ 
9    od
10 od
11 return  $\max_{i \in [m]} C_i$ 

```

- a) Wenden Sie den Algorithmus auf folgende Instanz an. Geben Sie dabei den Schedule am Ende des Algorithmus' an. Die Instanz hat $n = 4$ Jobs und $m = 3$ Maschinen, dabei sind die Jobs J durch folgende Tabelle gegeben.

Job	1	2	3	4
Ausführungszeit p	1	1	2	3
Maschinenbedarf q	1	3	2	1



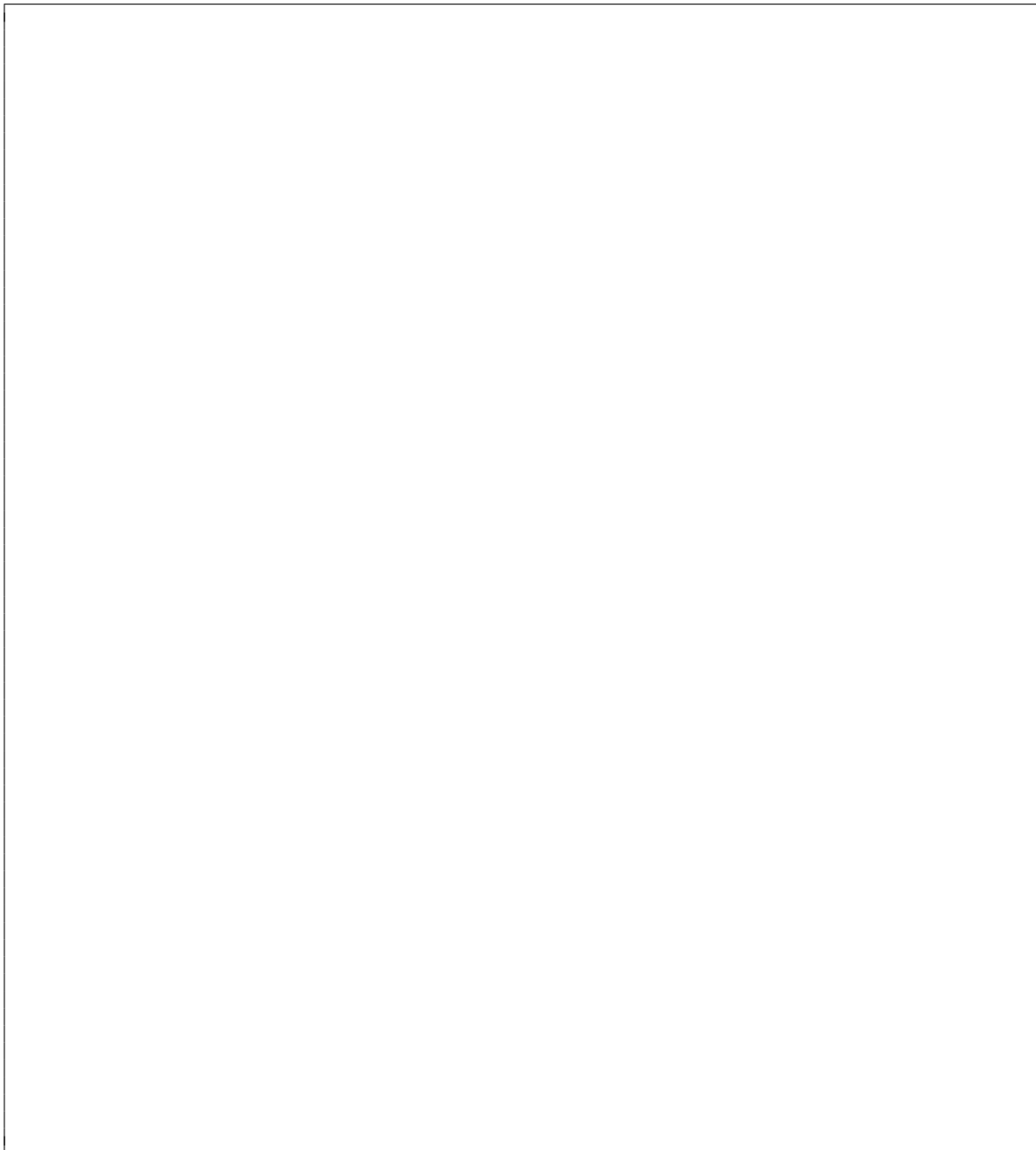
Matrikel-Nr.:

Name:

- b) Welche Worst-Case Güte hat der **LISTSCHEDULING** Algorithmus, wenn $q_j = 1$ für alle Jobs $j \in [n]$ gilt?




- c) Zeigen Sie $\text{LISTSCHEDULING}(I) \leq \text{OPT}(I) + \frac{1}{2}p_{\max}$ für alle Instanzen I mit $\frac{1}{3}m < q_j \leq \frac{1}{2}m$ für alle $j \in [n]$. Hier beschreibt $p_{\max} = \max_{j \in [n]} p_j$ die maximale Ausführungszeit in der Instanz.




d) **BONUS:** Sei $k \in \mathbb{N}_{\geq 2}$. Welche Güte hat **LISTSCHEDULING** für alle Instanzen I mit $\frac{1}{k+1}m < q_j \leq \frac{1}{k}m$ für alle $j \in [n]$? Beweisen Sie diese Güte.

Name: _____ Matrikel-Nr.: _____





CHRISTIAN-ALBRECHTS-UNIVERSITÄT ZU KIEL

Institut für Informatik, Arbeitsgruppe Algorithmen und Komplexität
Prof. Dr. K. Jansen

14. Oktober 2025

Modulprüfung zur Vorlesung »Analyse von Algorithmen und Komplexität – Komplexitätsteil«

SS 2025

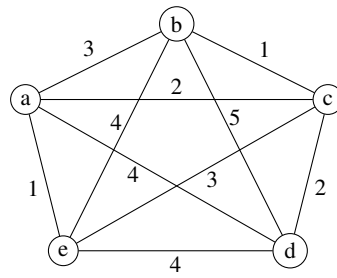
Matrikel-Nr.: _____

Hinweise:

- Bearbeiten Sie *alle* Aufgaben.
- Sie haben 10 Minuten Einlesezeit. Danach haben Sie **120 Minuten** Zeit für die Klausurbearbeitung.
- Erlaubte Hilfsmittel: Ein doppelseitig handschriftlich beschriebenes Blatt Papier DIN A4, (bunte) Stifte (kein Rot, kein Grün, kein Bleistift).
- Einsichtnahme: **6. November 2025, 09:30 bis 10:30 Uhr**. Der Raum für die Einsicht wird noch bekannt gegeben.
- Die Klausur besteht aus **5 Aufgaben**, die alle **10 Punkte** wert sind.
- Zum Bestehen müssen mindestens **18 Punkte** erreicht werden.
- Aufgabe 5 gibt zusätzlich **10 Bonuspunkte**. Insgesamt können also **50 Punkte plus 10 Bonuspunkte** erreicht werden.

Aufgabe 1 ANWENDUNG – ΔTSP_1 (5+1+2+2 Punkte)

Gegeben sei folgender Graph $G = (V, E)$:



- a) Wenden Sie den Algorithmus ΔTSP_1 auf den Graphen G mit **Startknoten a** an. Geben Sie dabei alle Graphen an, die in Zwischenschritten entstehen. Geben Sie außerdem die Länge der berechneten Rundreise an.

Matrikel-Nr.: _____



b) Welche Güte hat der Algorithmus ΔTSP_1 ?



c) Ist das Entscheidungsproblem zum TSP in P? Begründen Sie Ihre Antwort kurz.



d) Ist die folgende Aussage wahr? Begründen Sie Ihre Antwort kurz.

Wenn der Algorithmus ΔTSP_1 eine multiplikative Güte von 1,5 besitzt, dann bedeutet das, dass $\frac{d(R)}{OPT(I)} \geq 1,5$ für eine Instanz I und die Länge der vom Algorithmus für I berechneten Rundreise $d(R)$ gilt.



Aufgabe 2 VORLESUNGSBEWEIS – NP-VOLLSTÄNDIGKEIT (10 Punkte)

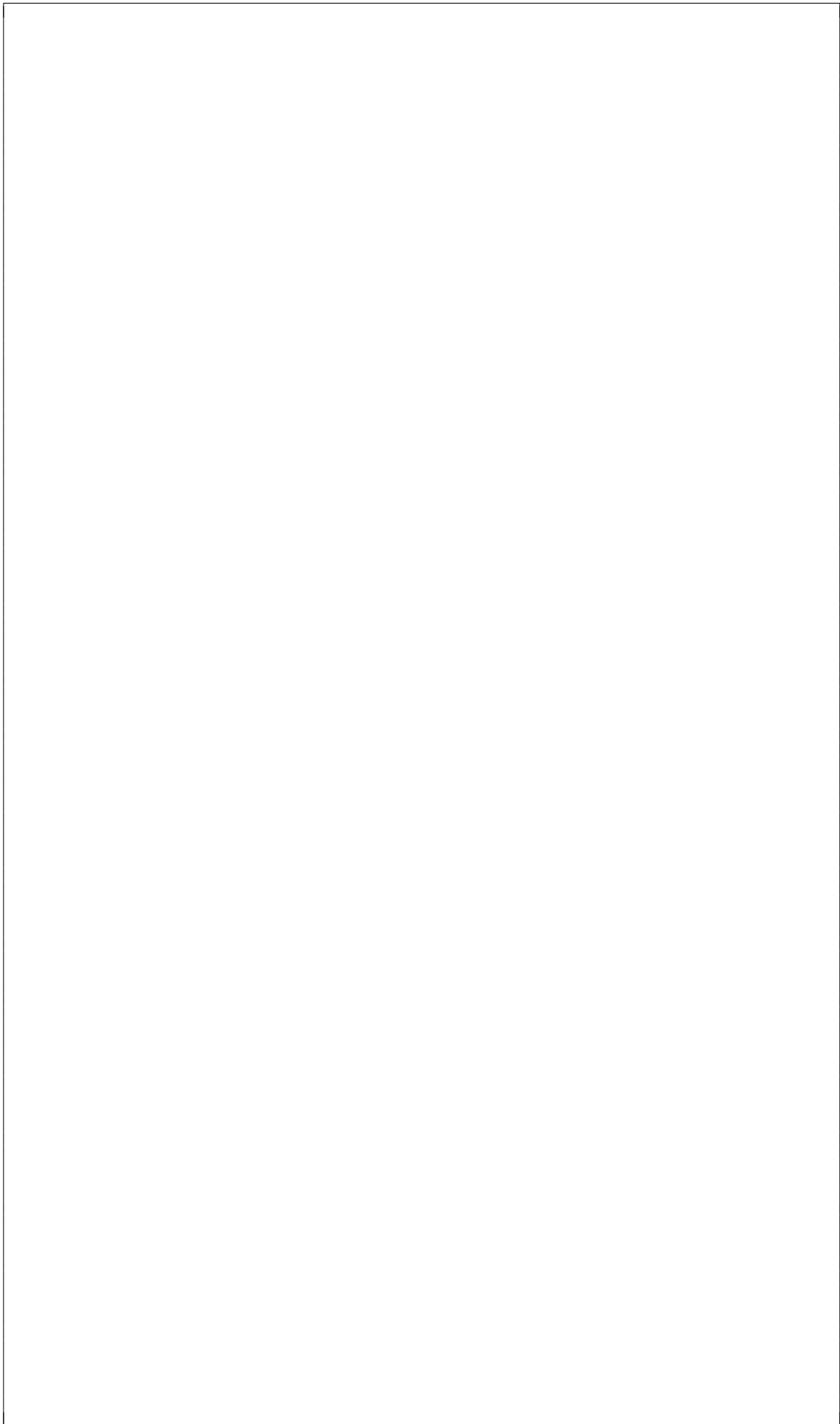
Beweisen Sie den folgenden aus der Vorlesung bekannten Satz.

Sei L_0 eine NP-vollständige Sprache. Falls $L_0 \in P$, dann gilt auch $P = NP$.

Matrikel-Nr.: _____



Matrikel-Nr.: _____



Aufgabe 3 NP-VOLLSTÄNDIGKEIT (10 Punkte)

Wir betrachten folgendes Problem:

Problem: k -COLOR-PRECOLORING

Eingabe: Ein ungerichteter Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}_{>0}$ sowie paarweise verschiedene Knoten $v_1, \dots, v_k \in V$.

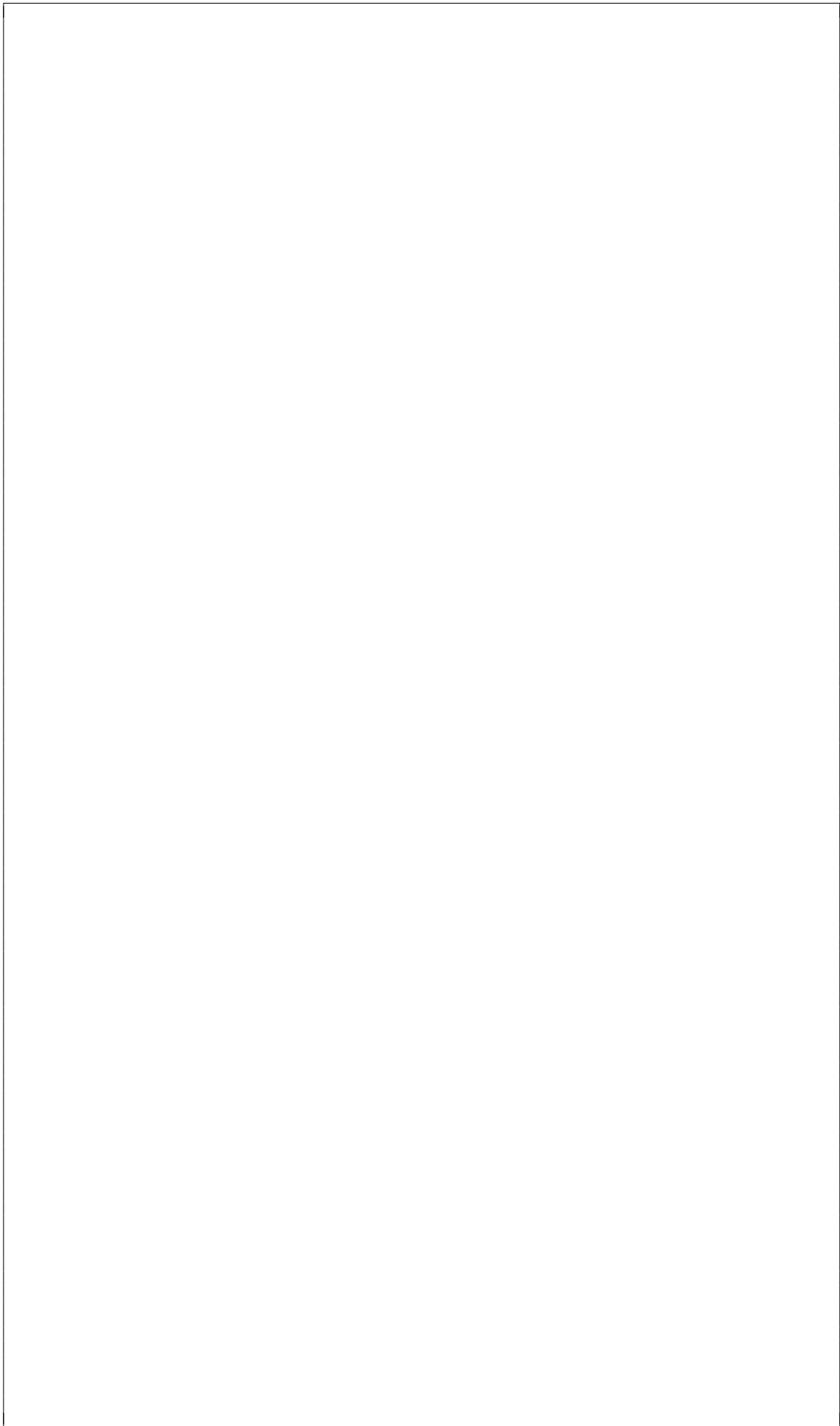
Entscheide: Existiert eine Färbung der Knoten $f : V \rightarrow [k]$, sodass $f(v_i) = i$ für alle $i \in [k]$ und $f(v) \neq f(u)$ für alle $\{v, u\} \in E$.

Zeigen Sie, dass k -COLOR-PRECOLORING NP-vollständig ist.

Matrikel-Nr.: _____



Matrikel-Nr.: _____



Aufgabe 4 ETH (1+4+1+4 Punkte)

Problem: COVERCLIQUE

Eingabe: Ein (ungerichteter) Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}_{\geq 1}$.

Entscheide: Existieren k paarweise disjunkte Cliques $C_1, \dots, C_k$, die alle Knoten überdecken.
Formeller: Existieren $C_1, \dots, C_k \subseteq V$, so dass $V = \bigcup_{i=1}^k C_i$ mit $\{\{v, u\} \mid v, u \in C_i, v \neq u\} \subseteq E$ für alle $i \in [k]$ und $C_i \cap C_j \neq \emptyset \implies i = j$ für alle $i, j \in [k]$.

Betrachten Sie folgende Reduktion von k -COLOR auf COVERCLIQUE: Für eine Eingabe $(G = (V, E), k)$ erstelle die Instanz $(G' = (V, \bar{E}), k)$, wobei $\bar{E}$ die invertierte Kantenmenge ist, also $\bar{E} = \{\{v, u\} \mid v, u \in V, v \neq u, \{v, u\} \notin E\}$.

- Geben Sie die Anzahl der Knoten und die Anzahl der Kanten für die resultierende COVERCLIQUE Instanz in Abhängigkeit von der originalen k -COLOR Instanz an.
- Beweisen Sie Laufzeit Lower Bounds für COVERCLIQUE bezüglich Anzahl der Knoten und Anzahl der Kanten basierend auf der ETH und der oben beschriebenen Reduktion.

Hinweis: k -COLOR kann unter Annahme der ETH nicht in $2^{o(|V|)} \cdot |I|^{O(1)}$ gelöst werden.

Matrikel-Nr.:

Problem: Δ COVER**Eingabe:** Ein (ungerichteter) Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}_0$.**Entscheide:** Besitzt der Graph eine Dreiecksüberdeckung $C_\Delta \subseteq V$ mit $|C_\Delta| \leq k$. C_Δ ist eine Dreiecksüberdeckung, wenn für alle $v, u, w \in V$ mit $\{v, u\}, \{v, w\}, \{u, w\} \in E$ gilt, dass $\{v, u, w\} \cap C_\Delta \neq \emptyset$.

Betrachten Sie folgende Reduktion von VERTEXCOVER auf Δ COVER bekannt aus der Übung: Für eine Instanz von VERTEXCOVER $(G = (V, E), k)$, bilde die Instanz $(G' = (V', E'), k)$ mit $V' = V \cup \{v_e \mid e \in E\}$ und $E' = E \cup \{\{v_e, e_1\}, \{v_e, e_2\} \mid e = \{e_1, e_2\} \in E\}$.

- c) Geben Sie die Anzahl der Knoten und die Anzahl der Kanten für die resultierende Δ COVER Instanz in Abhängigkeit von der originalen VERTEXCOVER Instanz an.
- d) Beweisen Sie Laufzeit Lower Bounds für Δ COVER bezüglich Anzahl der Knoten und Anzahl der Kanten basierend auf der ETH und der oben beschriebenen Reduktion.



Aufgabe 5 APPROXIMATIVE ALGORITHMEN - STRIP PACKING (2 + 3 + 5 + (10) Punkte)

Problem: STRIPPACKING

Eingabe: Eine Folge $L = (r_1, \dots, r_n)$ von Rechtecken mit Breite $w(r_i)$ und Höhe $h(r_i)$, so dass $h(r_i) \geq h(r_{i+1})$ für alle $i \in \{1, \dots, n-1\}$; ein Streifen mit Breite 1 und unbeschränkter Höhe.

Ausgabe: Eine Packung dieser Rechtecke in den Streifen so, dass sich keine zwei Rechtecke überlappen und die Höhe des gepackten Streifens minimal ist.

Der folgende Algorithmus für das StripPacking-Problem geht die Rechtecke nacheinander durch und packt sie von links nach rechts ganz unten auf den Streifen, bis kein Platz mehr ist. Dann geht er eine Stufe höher, also über das erste gepackte Rechteck ganz links, und packt auf dieser Höhe nach dem gleichen Prinzip weiter:

Algorithmus NEXT FIT DECREASING HEIGHT(NFDH)(L)

```

1  Stufe  $s \leftarrow 1$ 
2  foreach  $i \leftarrow 1, \dots, n$  do
3    Falls in Stufe  $s$  noch Platz ist, packe Rechteck  $r_i$  (links-justiert) auf
    diese Stufe.
4    Ansonsten, setze  $s \leftarrow s+1$  und packe dann Rechteck  $r_i$  (links-justiert)
    auf Stufe  $s$ .
5  od

```

a) Gegeben sei die folgende Beispielinstantz sowie eine Packung, erzeugt von NFDH. Geben Sie eine optimale Packung an sowie die Approximationsrate des Algorithmus' in diesem Fall.

Rechteck	1	2	3	4	5
Höhe ($h(r_i)$)	1	1	1	1	1
Breite ($w(r_i)$)	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{2}$

3	5	
2	3	4
1	1	2

Matrikel-Nr.:



b) Zeigen Sie: Es existieren Eingaben $L_n = (r_1, \dots, r_n)$ mit

$$\lim_{n \rightarrow \infty} \frac{\text{NFDH}(L_n)}{\text{OPT}(L_n)} = 2$$

Matrikel-Nr.: _____



c) Zeigen Sie, dass für jede Eingabeinstanz $L = (r_1, \dots, r_n)$ von Rechtecken mit $h(r_i) = 1$ gilt:

$$\text{NFDH}(L) \leq 2 \cdot \text{OPT}(L) + 1$$

Matrikel-Nr.: _____



- d) BONUS: Zeigen Sie, dass für jede Eingabeinstanz $L = (r_1, \dots, r_n)$ von Rechtecken mit $h(r_i) \leq 1$ gilt:

$$\text{NFDH}(L) \leq 2 \cdot \text{OPT}(L) + 1$$

Matrikel-Nr.: _____

