



CHRISTIAN-ALBRECHTS-UNIVERSITÄT ZU KIEL

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

19. Juli 2024

Modulprüfung zur Vorlesung »Analyse von Algorithmen und Komplexität (Komplexitätsteil)«

SS 2024

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, (bunte) Stifte (kein Rot, kein Grün, kein Bleistift).
- Einsichtnahme: 18.08.2024.

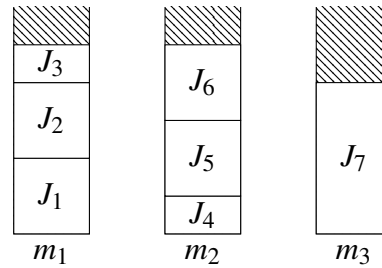
Note	4,0	3,7	3,3	3,0	2,7	2,3	2,0	1,7	1,3	1,0
Mindestpunktzahl	17,5	19,5	21,5	23,5	25	27	29	31	33	35

Aufgabe 1 KOMPLEXITÄT: ANWENDUNG LPT (4+2+2+2 Punkte)

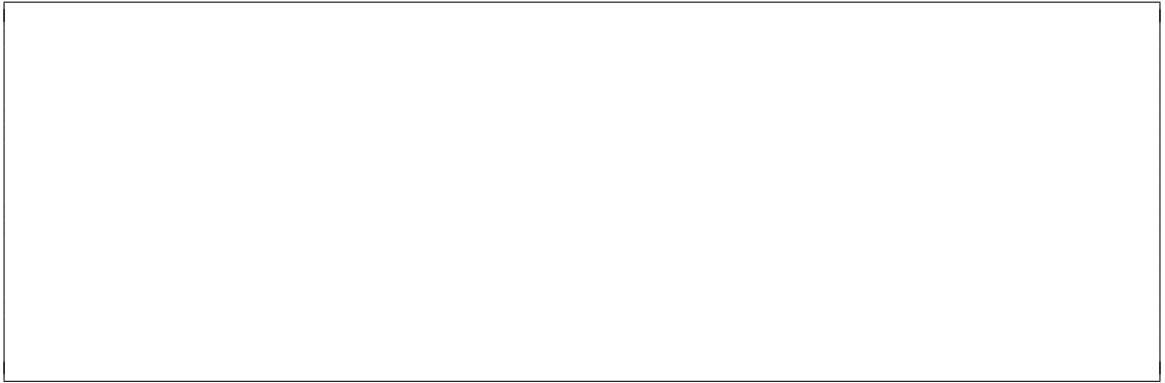
Gegeben ist folgende Scheduling Instanz:

J_j	J_1	J_2	J_3	J_4	J_5	J_6	J_7	
p_j	5	3	7	8	1	4	2	$m = 3$

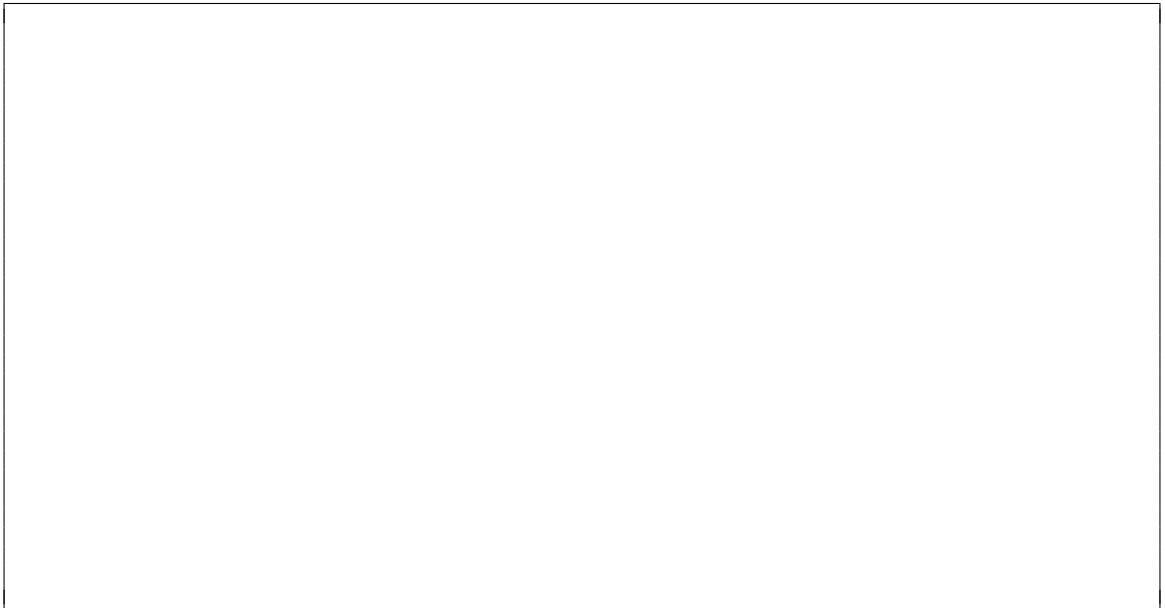
- a) Wenden Sie den LPT-Scheduling Algorithmus auf die obige Instanz an. Geben Sie dafür alle relevanten Zwischenschritte an, sowie die Reihenfolge in der die Jobs auf Maschinen platziert werden. Stellen Sie den resultierende Schedule graphisch dar. Eine beispielhafte Darstellung ist:



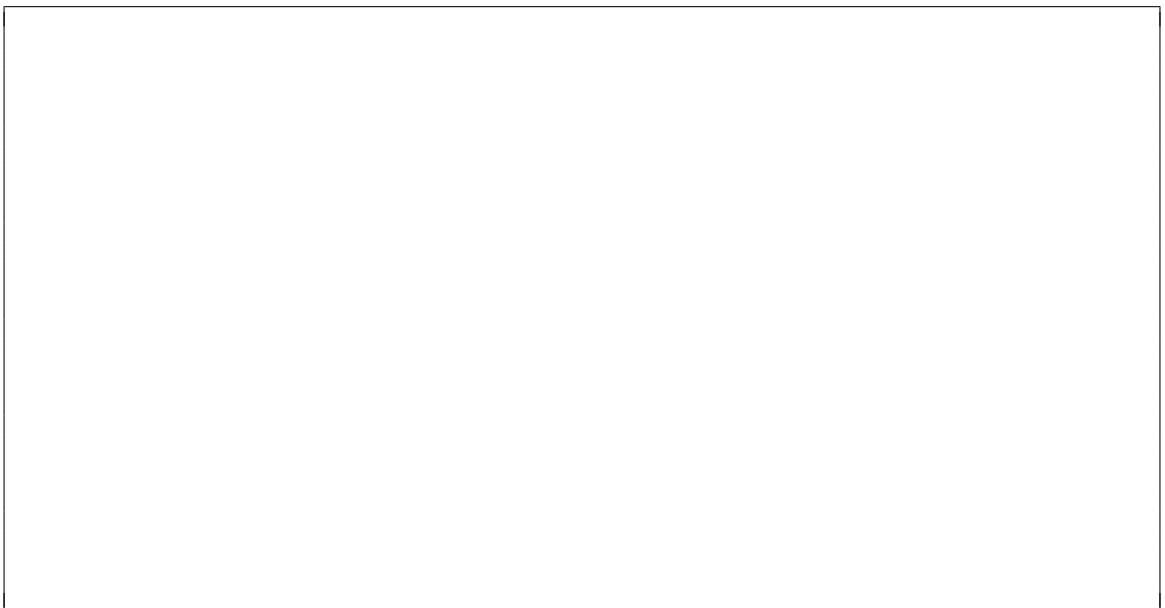
b) Geben Sie die approximative Güte des Algorithmus LPT an.



c) Geben Sie eine Instanz für $m = 2$ Maschinen an, bei denen der Algorithmus ListScheduling eine Güte von $2 - 1/m$ erzielt. Geben Sie zusätzlich an, welche Werte die optimale Lösung OPT und die Lösung von LPT für diese Instanz erzielen.



d) Geben Sie eine Instanz für $m = 3$ Maschinen an, bei denen der Algorithmus ListScheduling eine Güte von $2 - 1/m$ erzielt. Geben Sie zusätzlich an, welche Werte die optimale Lösung OPT und die Lösung von LPT für diese Instanz erzielen.



Aufgabe 2 KOMPLEXITÄT: BEWEIS: K-CLIQUE (10 Punkte)

Beweisen Sie folgende Aussage aus der Vorlesung:

$\text{SAT} \leq k\text{-Clique}$.

Hinweis: NICHT 3-SAT, sondern allgemeines SAT!



Wir betrachten folgendes Problem:

Problem: CliqueAndIndependentSet

Eingabe: Ein ungerichteter Graph $G = (V, E)$ mit $V = 1, \dots, |V|$ und eine ganze Zahl $k \geq 1$.

Entscheide: Gibt es in G sowohl ein Independent Set (Menge von Knoten, die paarweise nicht adjazent sind, d.h. zwischen denen es keine Kanten gibt) der Größe k und als auch eine Clique der Größe k ?

Zeigen Sie die NP-Schwere von CliqueAndIndependentSet durch Angabe einer Reduktion eines nach Vorlesung/Übung bekanntermaßen NP-schweren Problems.



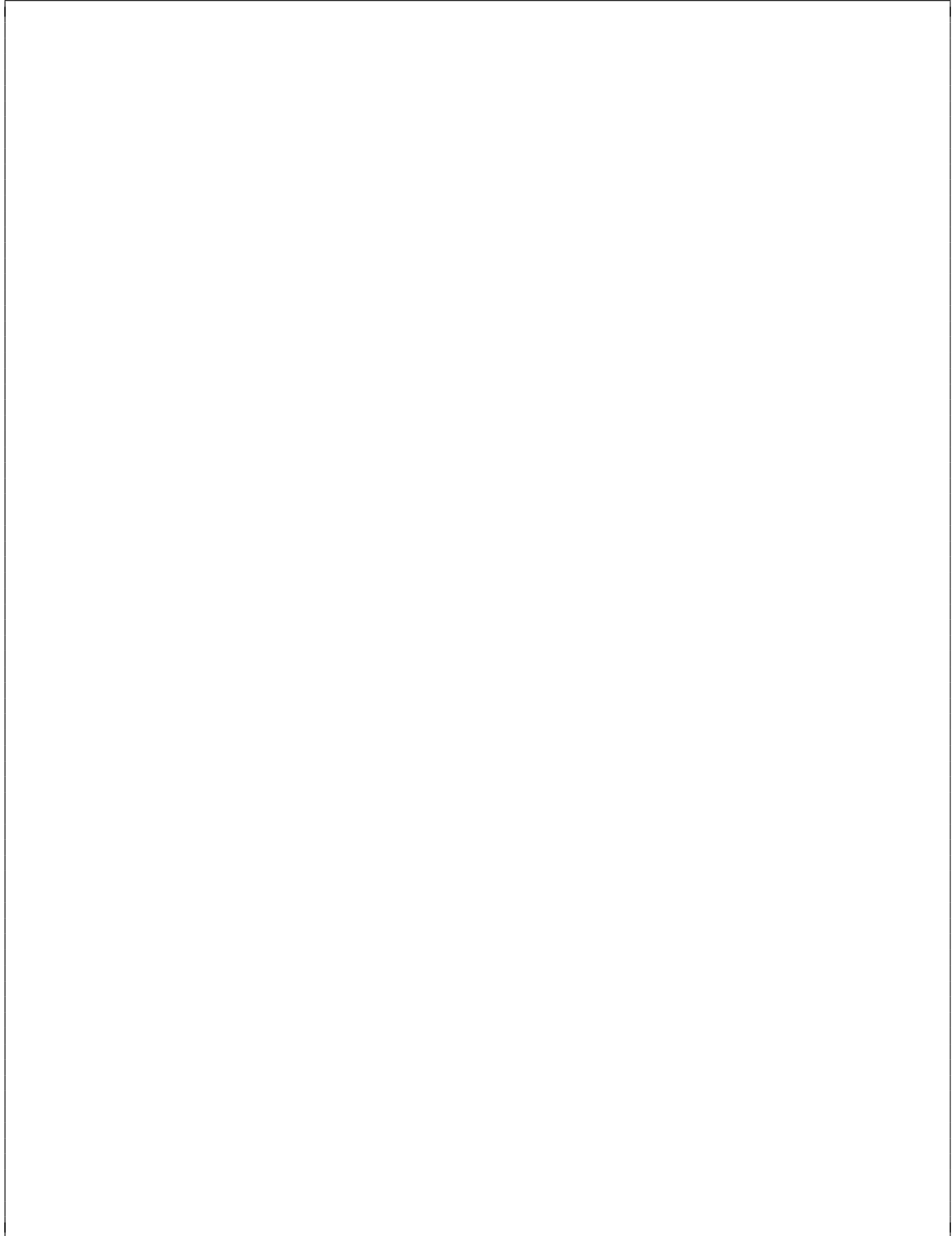
b) Wir betrachten folgendes Problem:

Problem: $(a_1 = 1)$ -SubsetSum

Eingabe: Eine Menge von n Items, jedes Item $i \in [n]$ hat eine Größe $a_i \in \mathbb{N}_{>0}$, wobei $a_1 = 1$, und ein Zielwert T .

Entscheide: Gibt es $I \subseteq [n]$ sodass $\sum_{i \in I} a_i = T$?

Zeigen Sie die NP-Schwere von $(a_1 = 1)$ -SubsetSum durch Angabe einer Reduktion eines nach Vorlesung/Übung bekanntermaßen NP-schweren Problems.



Aufgabe 4 KOMPLEXITÄT: APPROXIMATIVE ALGORITHMEN (3+7 Punkte)

Wir betrachten folgendes Problem:

Problem: APPROXIMATE SUBSET SUM

Eingabe: n ganze Zahlen $a_1, \dots, a_n$ und ein Zielwert T . Es gilt $a_i \leq T \quad \forall i \in [n]$ sowie $\sum_{i \in [n]} a_i > T$.

Ziel: Wähle Zahlenmenge $S \subseteq \{1, \dots, n\}$ mit $\sum_{j \in S} a_j \leq T$ und maximiere $\sum_{j \in S} a_j$.

Betrachten Sie zu diesem Problem den folgenden Algorithmus GA.

Algorithmus GA((A,T))

```
1  sortiere Zahlen absteigend, sodass  $a_1 \geq a_2 \geq \dots \geq a_n$  gilt;  
2  integer  $i = 1$ ;  
3  integer  $sum = 0$ ;  
4  boolean  $tooLarge = false$ ;  
5  while  $i \leq n$  and  $tooLarge = false$  do  
6    if  $sum + a_i \leq T$  then  
7       $sum = sum + a_i$ ;  
8       $i = i + 1$ ;  
9    else  
10      $tooLarge = true$ ;  
11  fi  
12 od  
13 return  $sum$ ;
```

- a) Geben Sie eine Konstruktionsvorschrift für eine Instanz I an, mit der der Algorithmus GA beliebig nah an eine Güte von 2 kommt, also $A(I) \approx \frac{OPT(I)}{2}$.



- b) Zeigen Sie, dass der Algorithmus GA eine approximative Güte von 2 hat, also $A(I) \geq \frac{OPT(I)}{2}$ gilt.



Aufgabe 5 KOMPLEXITÄT: ETH (5+5 Punkte)

a) Betrachten Sie nachfolgende Reduktion von 3-SAT auf k -Color:

1. Eliminiere doppelte Klauseln und Variablen, die in keiner Klausel auftauchen
2. Elimiere Klauseln die eine Variable und ihre Negation enthalten (diese sind immer erfüllt)
3. Seien $C_1, \dots, C_m$ die Klauseln in der SAT Formel
4. Definiere

$$V := \{x_i, \bar{x}_i, v_i \mid i \in [n]\} \cup \{C_j \mid j \in [m]\} \cup \{z\} ,$$

sowie

$$\begin{aligned} E := & \{ \{v_i, v_j\}, \{v_i, x_j\}, \{v_i, \bar{x}_j\} \mid i \in [n], j \in [n] \setminus \{i\} \} \\ & \cup \{ \{x_i, \bar{x}_i\} \mid i \in [n] \} \\ & \cup \{ \{x_i, C_j\} \mid i \in [n], j \in [m], x_i \notin C_j \} \\ & \cup \{ \{\bar{x}_i, C_j\} \mid i \in [n], j \in [m], \bar{x}_i \notin C_j \} \\ & \cup \{ \{v_i, z\} \mid i \in [n] \} \\ & \cup \{ \{C_j, z\} \mid j \in [m] \} \end{aligned}$$

Welche Lower Bounds ergeben sich unter Annahme der ETH for k -Colour aus dieser Reduktion in Hinblick auf

- (i) die Anzahl der Knoten $|V|$; sowie
- (ii) die Anzahl der Kanten $|E|$?



b) Wir betrachten folgendes Scheduling Problem:

Problem: $2|\text{prec}, p_i \in \{1, 2\}|C_{\max}$

Eingabe: Eine Menge $\mathcal{J}$ von n Jobs, jeder Job $J \in \mathcal{J}$ hat eine Ausführungszeit $p_J \in \{1, 2\}$ sowie Präzedenzconstraints in Form eines gerichteten, azyklischen Graphen $G = (\mathcal{J}, A)$, sowie ein ganze Zahl $T \geq 0$.

Entscheide: Existiert ein Schedule, d.h. eine Abbildung, die jedem Job eine Startzeit und eine Maschine zuweist, sodass (a) sich Jobs auf einer Maschine nicht überlappen (b) für alle Jobs $j, k \in \mathcal{J}$ mit $(j, k) \in A$, dass k erst startet nachdem j vollständig abgearbeitet ist und (c) der Makespan höchstens T ist?

Wir betrachten die folgende Reduktion von k -Clique auf $2|\text{prec}, p_i \in \{1, 2\}|C_{\max}$.

Seien $G = (V, E)$ und k gegeben. Wir setzen $n = 4|V| + 3|E|$ und erstellen einen Knoten-Job J_i für alle $i \in V$ mit $p_i = 1$. Des Weiteren erstellen wir Kanten-Jobs $J_{\{i, j\}}$ für jede Kante $\{i, j\} \in E$ mit $p_{\{i, j\}} = 2$. Abschließend erstellen wir noch $3|V| + 2|E|$ Dummy Jobs. Diese Jobs haben Ausführungszeit $p_{\text{dummy}} = 1$.

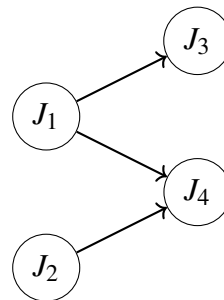
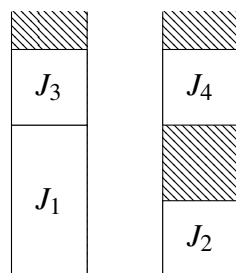
Wir setzen unsere Präzedenzen nach dem folgenden Schema:

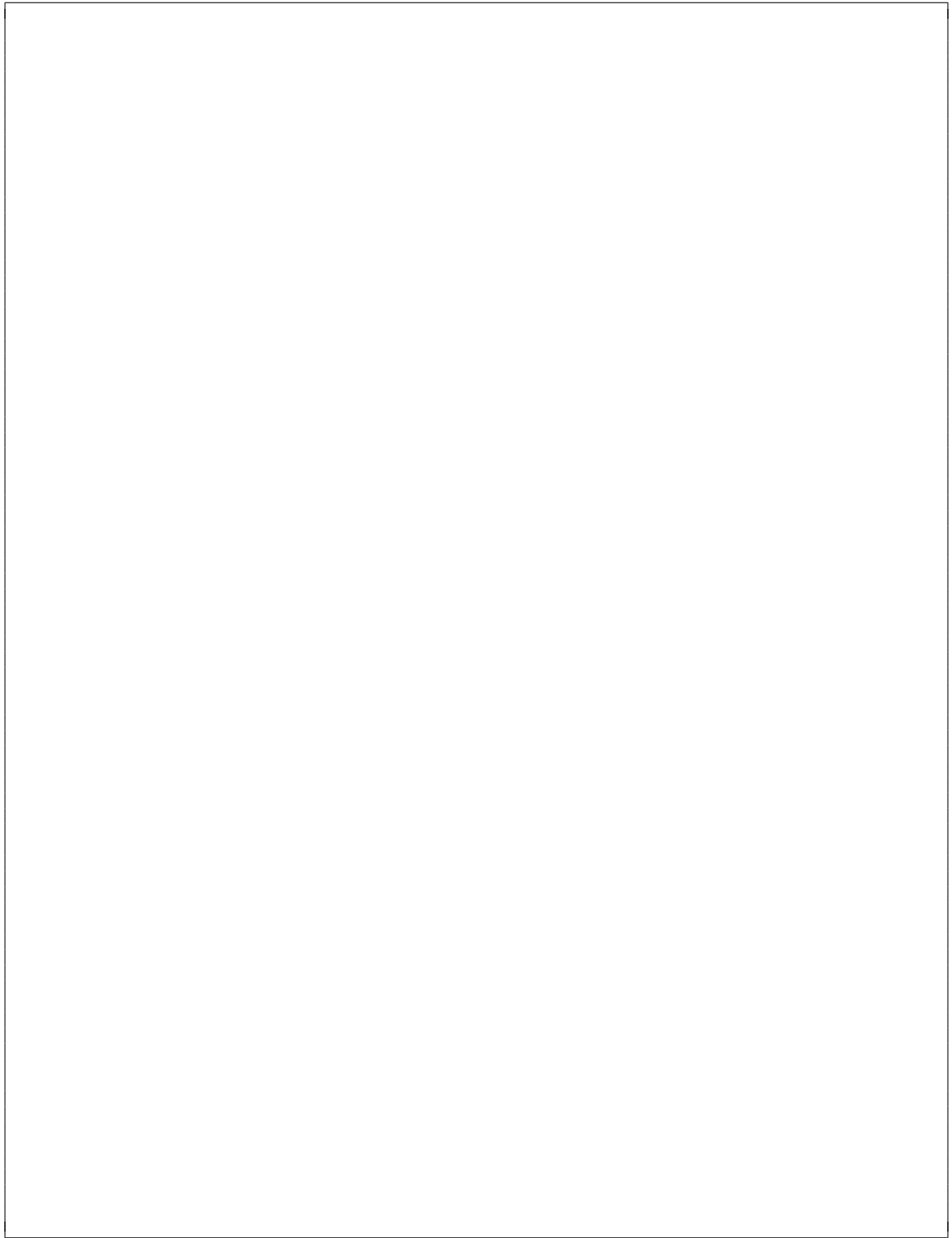
- 1) $(J_i, J_{\{i, j\}})$ für alle $i \in V, \{i, j\} \in E$
- 2) Präzedenzen innerhalb der Dummy Jobs.

Zuletzt setzen wir den Makespan $T = 2|V| + 2|E|$.

- (i) Geben Sie die Lower Bounds für $2|\text{prec}, p_i \in \{1, 2\}|C_{\max}$ in Hinblick auf n an, die sich unter der ETH aus dieser Reduktion, basierend auf den Lower Bounds für k -Clique, ergeben.
- (ii) Geben Sie die Lower Bounds für $2|\text{prec}, p_i \in \{1, 2\}|C_{\max}$ in Hinblick auf T an, die sich unter der ETH aus dieser Reduktion, basierend auf den Lower Bounds für k -Clique, ergeben.

Hinweis: Ein beispielhafter Schedule ist unten dargestellt. Die Präzedenzen sind im Graphen durch gerichtete Kanten abgebildet, daher startet J_4 erst nach dem Ende von J_1 . Nutzen Sie für diese Aufgabe die bekannten Lower Bounds aus der Präsenzübung. Diese wurden aus der Reduktion von 3-SAT auf k -Clique hergeleitet. Hierbei ist der Clique Graph zusammenhängend!

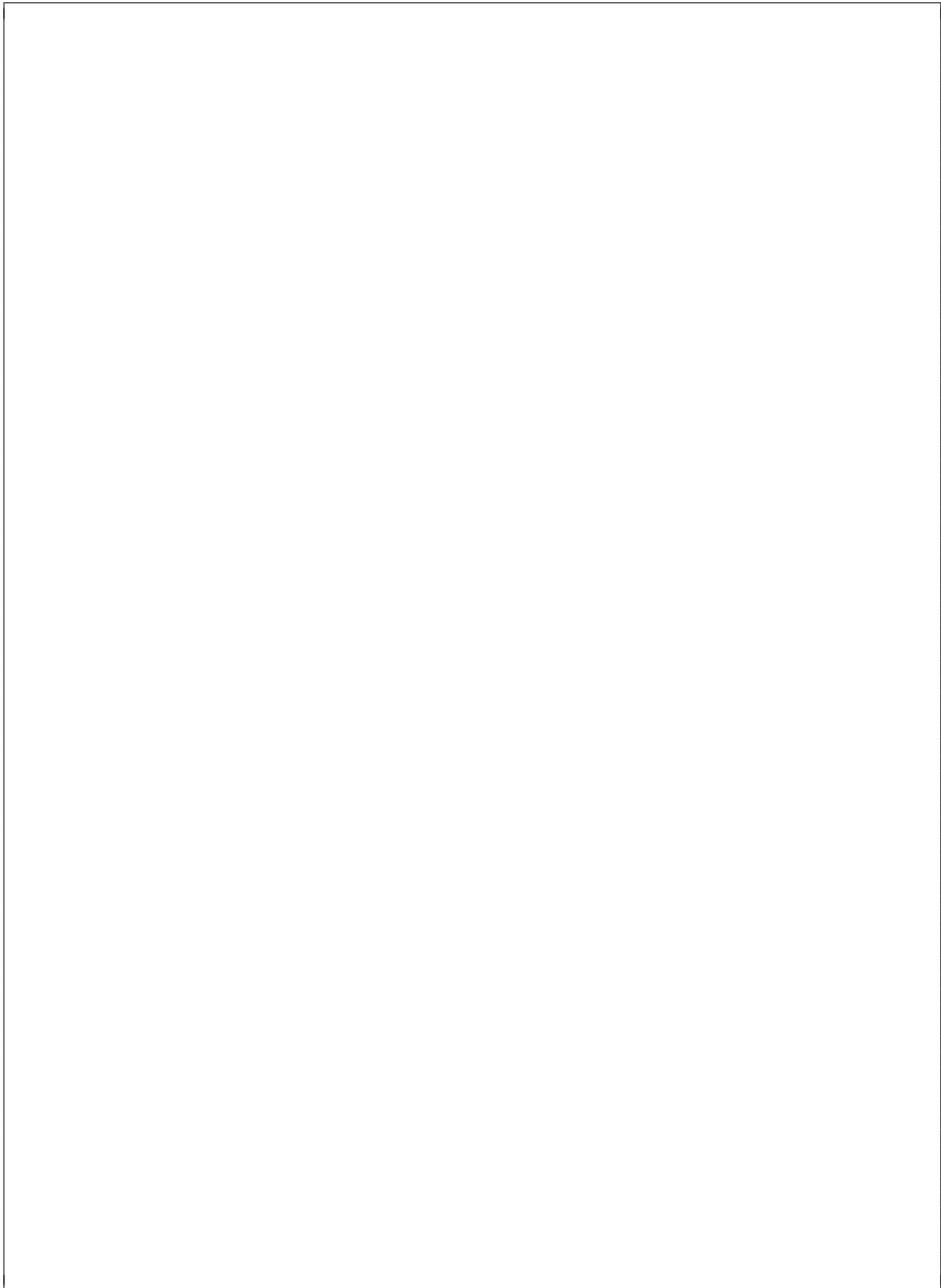




Aufgabe 6 BONUSAUFGABE (10 Punkte)

Beweisen oder widerlegen Sie folgende Aussage:

Wenn die ETH fehlschlägt gilt $P = NP$.



Viel Erfolg!



CHRISTIAN-ALBRECHTS-UNIVERSITÄT ZU KIEL

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

11. Oktober 2024

Modulprüfung zur Vorlesung »Analyse von Algorithmen und Komplexität (Komplexitätsteil)«

SS 2024

Name: _____ Matrikel-Nr.: _____

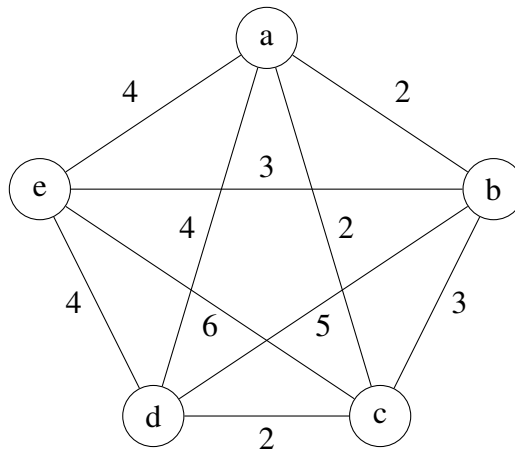
Hinweise:

- 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, (bunte) Stifte (kein Rot, kein Grün, kein Bleistift).
- Einsichtnahme: 30.10.2024

Note	4,0	3,7	3,3	3,0	2,7	2,3	2,0	1,7	1,3	1,0
Mindestpunktzahl	17,5	19,5	21,5	23,5	25	27	29	31	33	35

Aufgabe 1 ΔTSP_1 (6+2+2 Punkte)

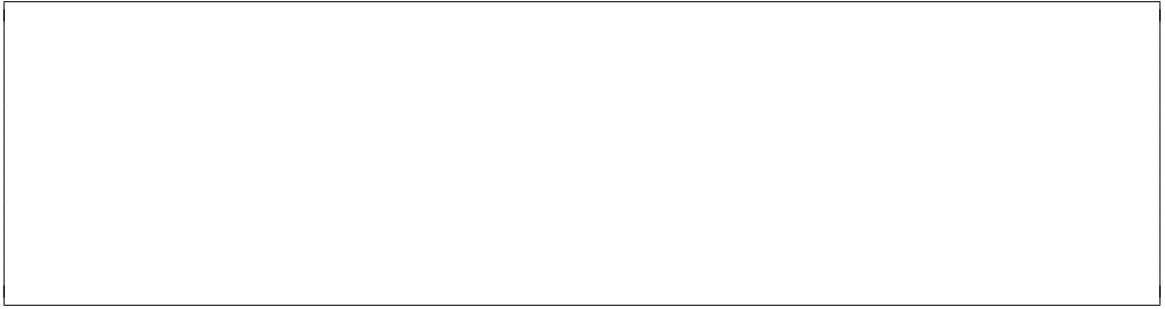
Gegeben sei folgender Graph $G = (V, E)$, bei dem die Kantengewichte jeweils an der nach außen gerichteten Seite der Kante stehen:



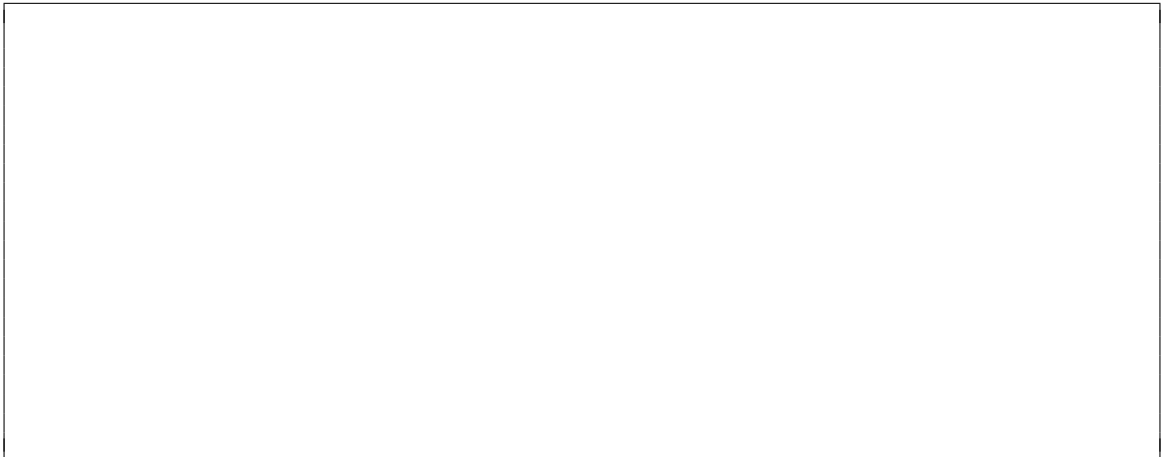
- a) Wenden Sie den Algorithmus ΔTSP_1 auf den Graphen G beginnend bei a an. Geben Sie dabei die Ergebnisse jedes Zwischenschrittes des Algorithmus an.



- b) Beschreiben Sie, was der Algorithmus ΔTSP_1 berechnet und geben Sie die Bedingungen an, die an den Graphen gestellt sind, damit der Algorithmus eine korrekte Lösung berechnet.



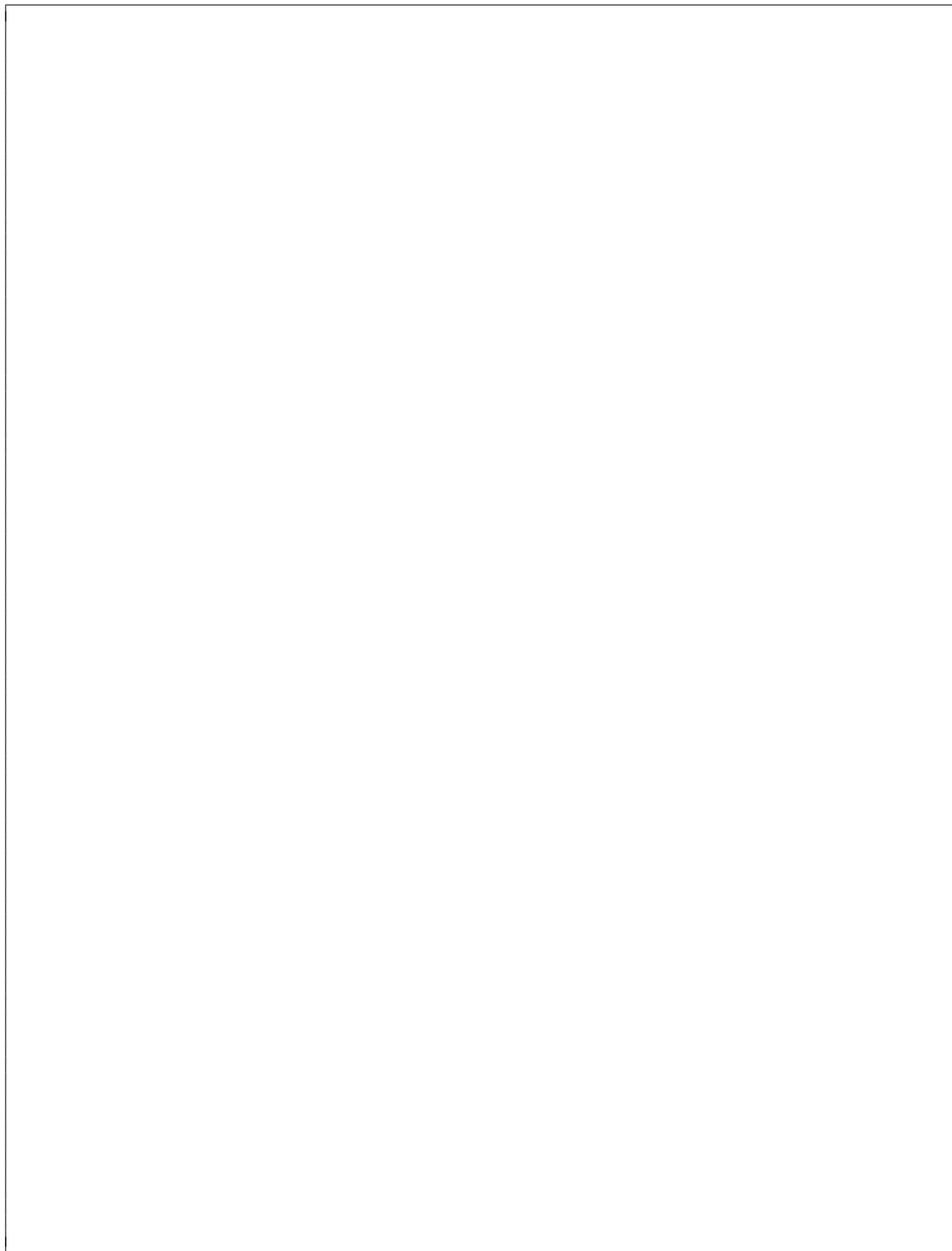
- c) Begründen Sie, warum der Algorithmus ΔTSP_1 eine approximative Güte von 2 besitzt.



Aufgabe 2 KOMPLEXITÄT: VORLESUNGSBEWEIS SAT (10 Punkte)

Beweisen Sie folgende Aussage:

3-SAT ist *NP*-vollständig. Zeigen Sie dafür $\text{SAT} \leq 3\text{-SAT}$, nehmen Sie also $3\text{-SAT} \in \text{NP}$ als gegeben an.



Aufgabe 3 KOMPLEXITÄT: NP-VOLLSTÄNDIGKEIT (5+5 Punkte)

a) Wir betrachten folgendes Problem:

Problem: Hitchhiker's-HamiltonianCycle

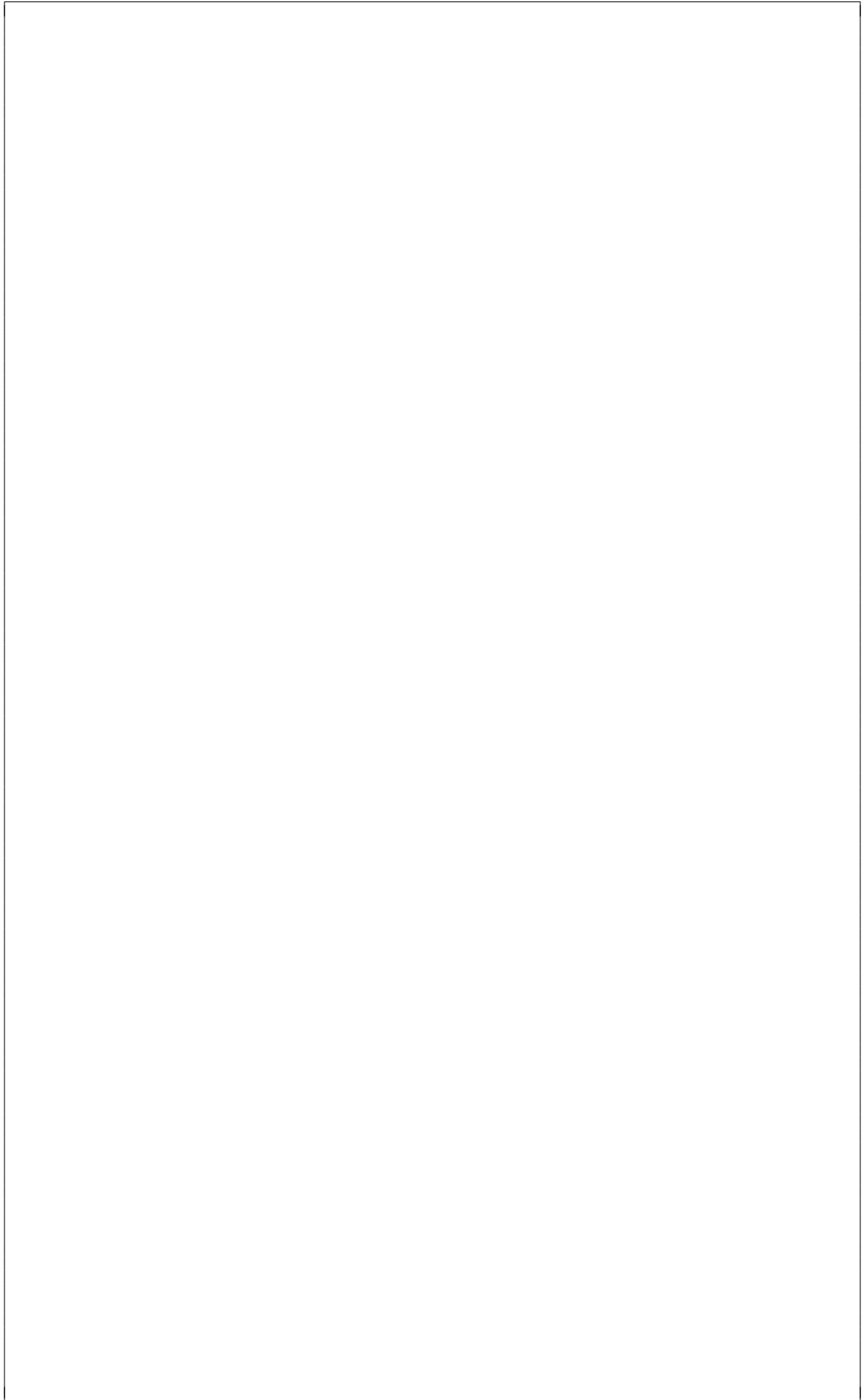
Eingabe: Ein ungerichteter Graph $G = (V, E)$ mit $V = \{1, \dots, |V|\}$ und für alle $v \in V$ gilt $\deg(v) \geq 42$.

Entscheide: Gibt es einen Hamiltonkreis in G ?

Beweisen Sie die NP-Schwere von Hitchhiker's-HamiltonianCycle durch Angabe einer Reduktion eines nach Vorlesung/Übung bekanntermaßen NP-schweren Problems.

Hinweis: Als Hilfe, geben Sie graphisch die Reduktion für Knotengrad ≥ 10 an.





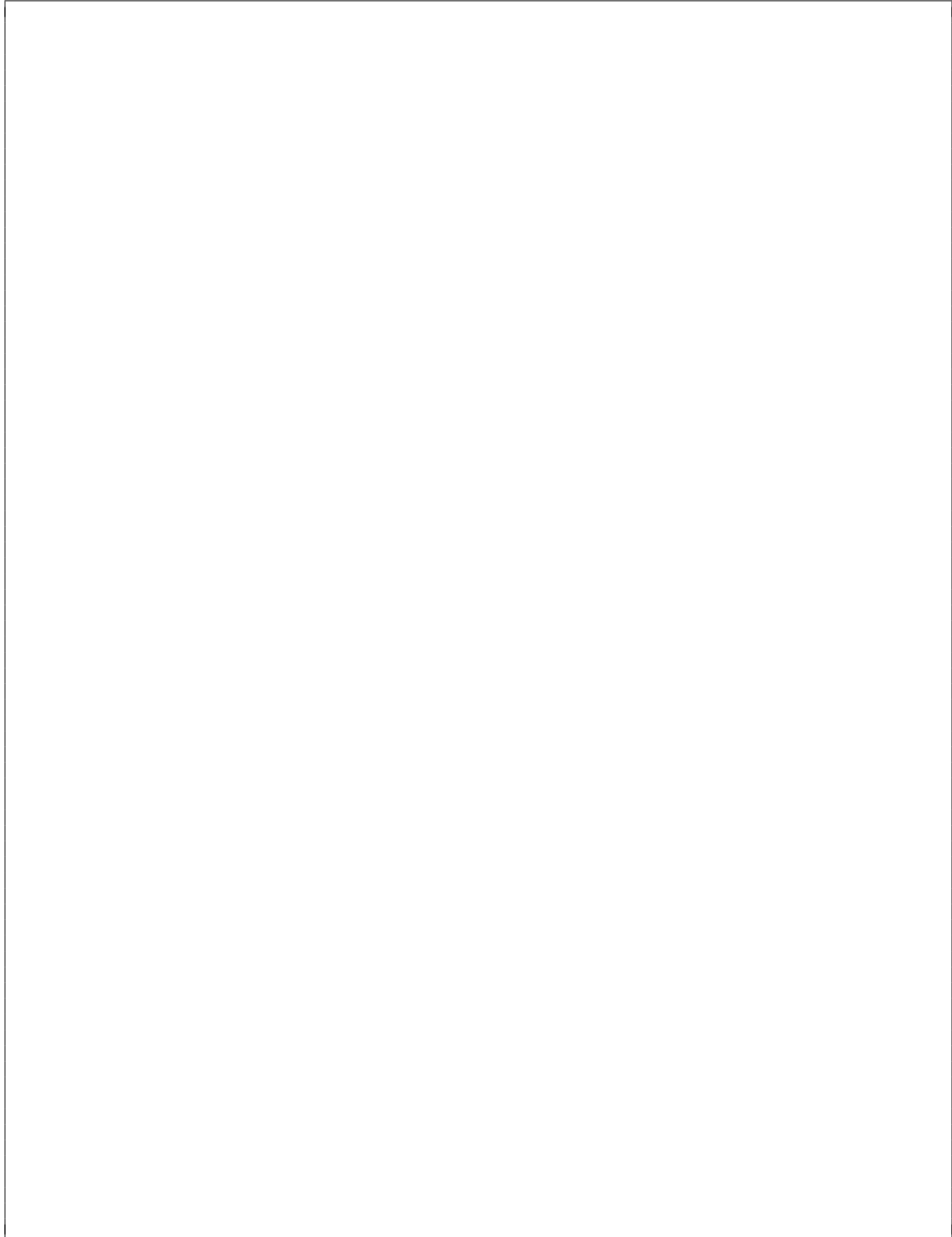
b) Wir betrachten folgendes Problem:

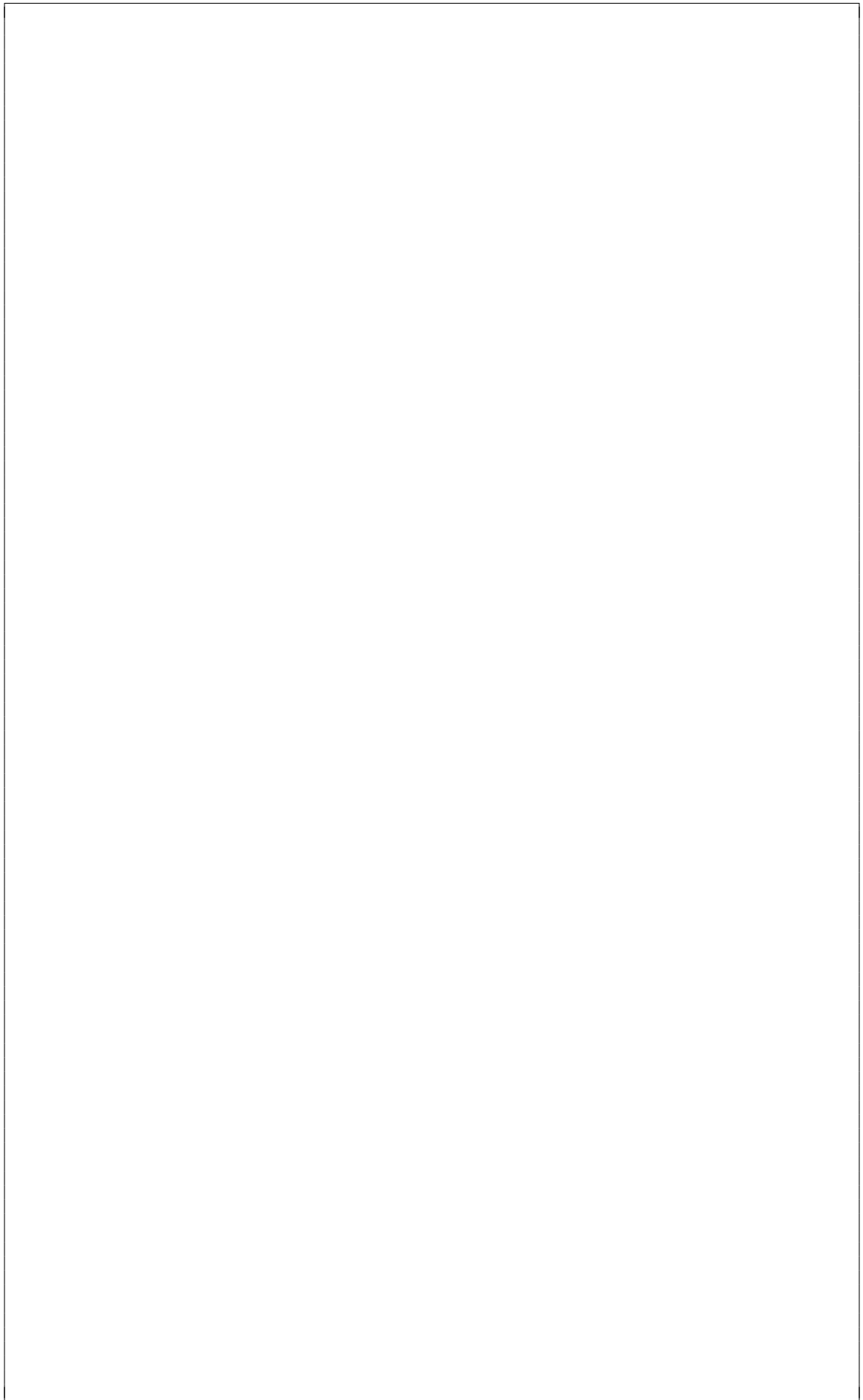
Problem: AtMostTwoPerSize-SubsetSum

Eingabe: Ein ganzzahliger Zielwert $T > 0$, eine Menge von n Items, jedes Item $i \in [n]$ hat eine Größe $a_i \in \mathbb{N}_{>0}$, jede Größe tritt höchstens zwei Mal auf, d.h. für alle $i \in [n]$ gilt $|\{i' \in [n] \mid a_{i'} = a_i\}| \leq 2$.

Entscheide: Gibt es $S \subseteq [n]$ sodass $\sum_{i \in S} a_i = T$?

Beweisen Sie die NP-Schwere von AtMostTwoPerSize-SubsetSum durch Angabe einer Reduktion eines nach Vorlesung/Übung bekanntermaßen NP-schweren Problems.





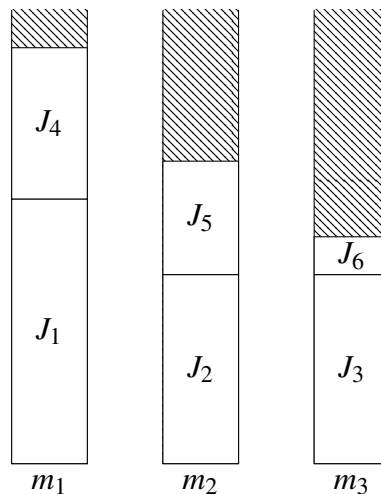
Aufgabe 4 KOMPLEXITÄT: APPROXIMATIVE ALGORITHMEN (10+(10) Punkte)

Wir betrachten das aus der Vorlesung bekannte Makespan Scheduling Problem $P||C_{\max}$. Dazu ist folgender Algorithmus gegeben:

Algorithmus ROUND ROBIN SCHEDULING ($I=(J,m)$)

```
1  sortiere die Jobs in  $J$  so, dass  $p_1 \geq p_2 \geq \dots \geq p_n$  gilt;  
2  setze  $B_1 = \dots = B_m = \emptyset$ ;  
3  integer  $j = 1$ ;  
4  integer  $i = 1$ ;  
5  while  $j \leq n$  do  
6    Platziere Job  $J_j$  auf Maschine  $M_i$ ;  
7     $B_i = B_i \cup \{J_j\}$ ;  
8     $j = j + 1$ ;  
9    if  $i < m$  then  
10      $i = i + 1$   
11  else  
12      $i = 1$   
13  fi  
14 od  
15 return  $B_1, \dots, B_m$ 
```

Betrachten Sie folgendes Beispiel zu der Funktionsweise des Round Robin Scheduling Algorithmus'. Gegeben sind 3 Maschinen und 6 Jobs $J_1, \dots, J_6$, die bereits richtig sortiert sind. Es gilt $p_1 = 7, p_2 = 5, p_3 = 5, p_4 = 4, p_5 = 3, p_6 = 1$. Daraus ergibt sich folgender Schedule:



- a) Zeigen Sie per Widerspruch, dass der Algorithmus Round Robin Scheduling eine approximative Güte von 2 hat.



- b) BONUS: Geben Sie die Konstruktionsvorschrift für eine Instanz an, mit der der Algorithmus Round Robin Scheduling beliebig nah an eine Güte von 2 kommt, also $A(I) \approx 2 \cdot OPT(I)$. Nutzen Sie dafür mindestens 3 Maschinen und 7 Jobs, womit eine Rate von $A(I) = \frac{5}{3} OPT$ zu erreichen ist. Geben sie die optimale Makespan an und die, die der Algorithmus bei Ihrer Instanz erreicht. Für eine Güte von $\frac{5}{3}$ können bis zu 3 Punkten erreicht werden, bei einer Güte von $\frac{7}{4}$ sind bis zu 5 Punkte zu erreichen. Geben Sie eine Konstruktionsvorschrift an, bei denen bei einer beliebigen, festen Anzahl von Maschinen m eine Güte von $2 - \frac{1}{m}$ erreicht wird, können Sie die vollen 10 Punkte erreichen.



Aufgabe 5 KOMPLEXITÄT: ETH (5+5 Punkte)

a) Wir betrachten folgendes Problem:

Problem: DominatingSet

Eingabe: Ein ungerichteter Graph $G = (V, E)$ mit $V = 1, \dots, |V|$ und eine Ganzzahl $k \geq 1$.

Entscheide: Gibt es eine Menge $M \subseteq V$ der Kardinalität $|M| = k$ derart, dass jeder Knoten $v \in V$ entweder in M ist oder zu einem Knoten in M benachbart ist.

Wir untersuchen nun folgende Reduktion von 3-SAT auf DominatingSet:

1. Seien $C_1, \dots, C_m$ die Klauseln der 3-SAT-Formel
2. Seien $x_1, \dots, x_n$ die Variablen der 3-SAT-Formel
3. Erzeuge die Knoten $x_i, \bar{x}_i$ und d_i für jede Variable x_i
4. Erzeuge die Kanten $\{x_i, \bar{x}_i\}$, $\{\bar{x}_i, d_i\}$ und $\{d_i, x_i\}$ für jede Variable x_i
5. Für jede Klausel C_j erzeuge einen Knoten C_j
6. Für jedes Literal ℓ einer Klausel C_j erzeuge die Kanten $\{C_j, \ell\}$

Welche unteren Schranken für die Laufzeit folgen damit für DominatingSet unter der ETH in Bezug auf

- (i) die Anzahl der Knoten $|V|$?
- (ii) die Anzahl der Kanten $|E|$?



b) Wir betrachten folgendes Problem:

Problem: GridTiling

Eingabe: Zwei ganze Zahlen k und j und für jedes $i \in [k]$ und $j \in [k]$ eine Menge $S_{i,j} \subseteq [n]^2$.

Entscheide: Gibt es $(e_{i,j})_{(i,j) \in [k]^2}$ derart, dass

1. für alle $(i,j) \in [k]^2$ gilt $e_{i,j} \in S_{i,j}$;
2. für alle $i \in [k]$ und $j \in [k-1]$ gilt für $e_{i,j} = (a,b)$ und $e_{i,j+1} = (a',b')$, dass $a = a'$ (dh. die ausgewählten Tupel müssen in der ersten Komponente übereinstimmen, wenn der erste Index derselbe ist); und
3. für alle $i \in [k-1]$ und $j \in [k]$ gilt für $e_{i,j} = (a,b)$ und $e_{i+1,j} = (a',b')$, dass $b = b'$ (dh. die ausgewählten Tupel müssen in der zweiten Komponente übereinstimmen, wenn der zweite Index derselbe ist)

Betrachten Sie folgende Beispielinstantz für $k = 2$, die Elemente der Lösung sind unterstrichen:

$S_{1,1} = \{(1, \underline{1}), (2, 2)\}$	$S_{1,2} = \{(\underline{1}, 2), (2, 1), (2, 2)\}$
$S_{2,1} = \{(1, 2), (2, \underline{1})\}$	$S_{2,2} = \{(1, 1), (\underline{2}, 2)\}$

Wir untersuchen folgende Reduktion von Clique:

1. Sei $G = (V, E)$ und eine Zahl k als zu untersuchende Clique Instanz gegeben
2. Wir nehmen ohne Beschränkung der Allgemeinheit an, dass $V = \{1, \dots, |V|\}$ sowie $k \geq 2$ gilt und dass keine isolierten Knoten existieren.
3. Wir übernehmen den Wert von k
4. Für jedes Paar $(i, j) \in [k]^2$ definieren wir

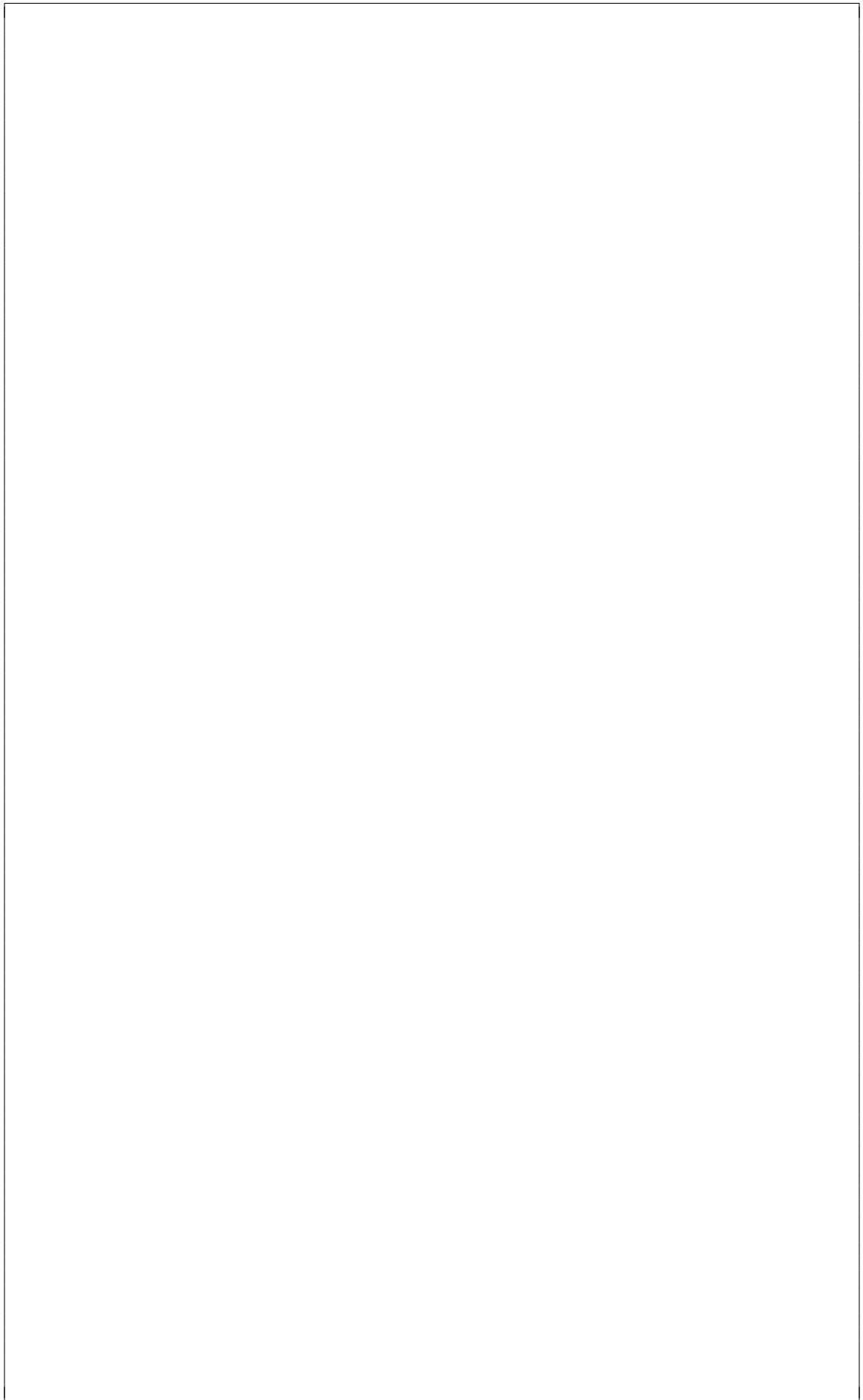
$$S_{i,j} := \begin{cases} \{(a, a) \mid a \in V\} & \text{if } i = j \\ \{(a, b) \mid a \neq b, \{a, b\} \in E\} & \text{if } i \neq j \end{cases}$$

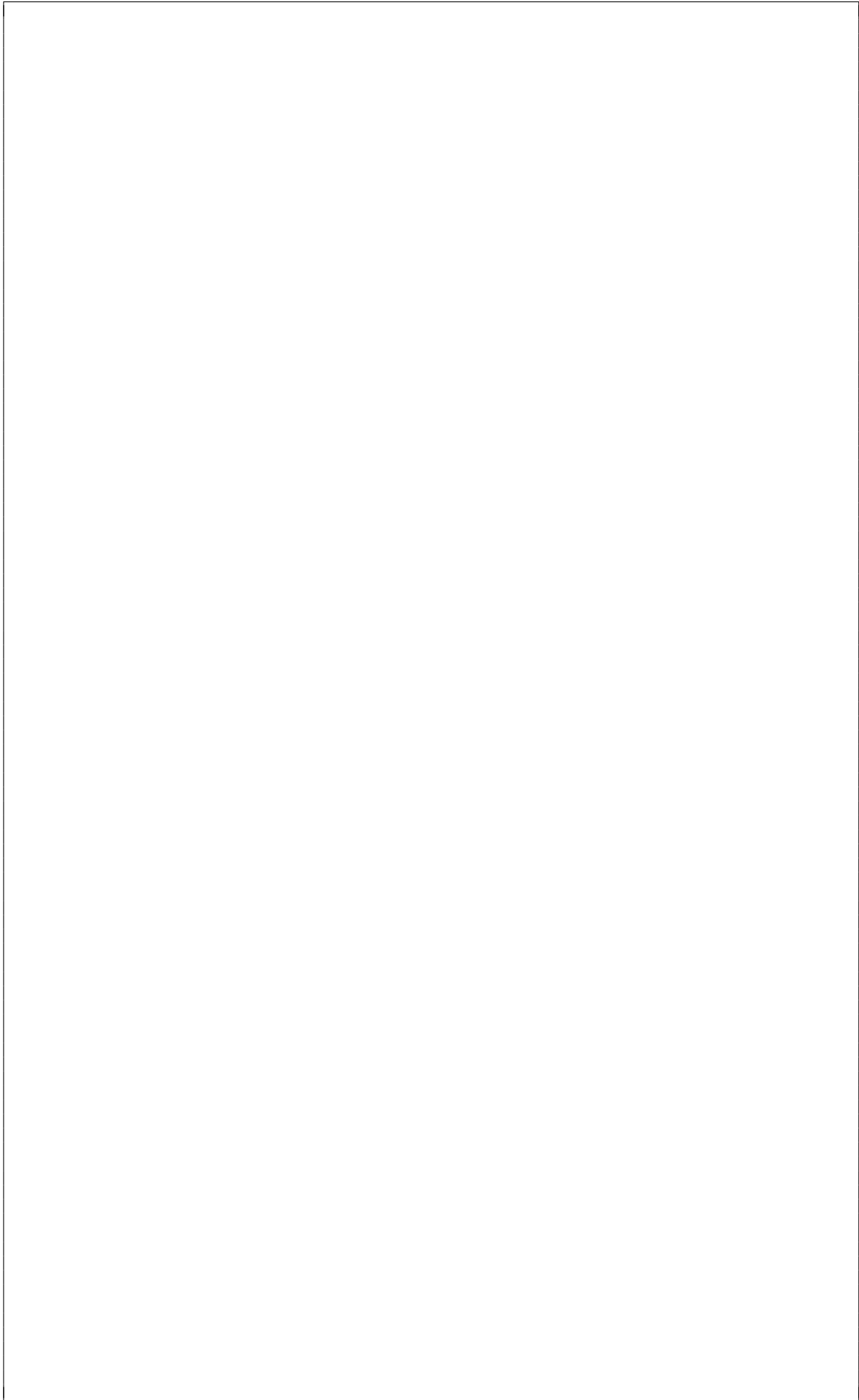
Welche unteren Schranken für die Laufzeit folgen damit für Grid Tiling (mit der aus der Vorlesung bekannten Reduktion von Clique auf 3-SAT) in Hinblick auf

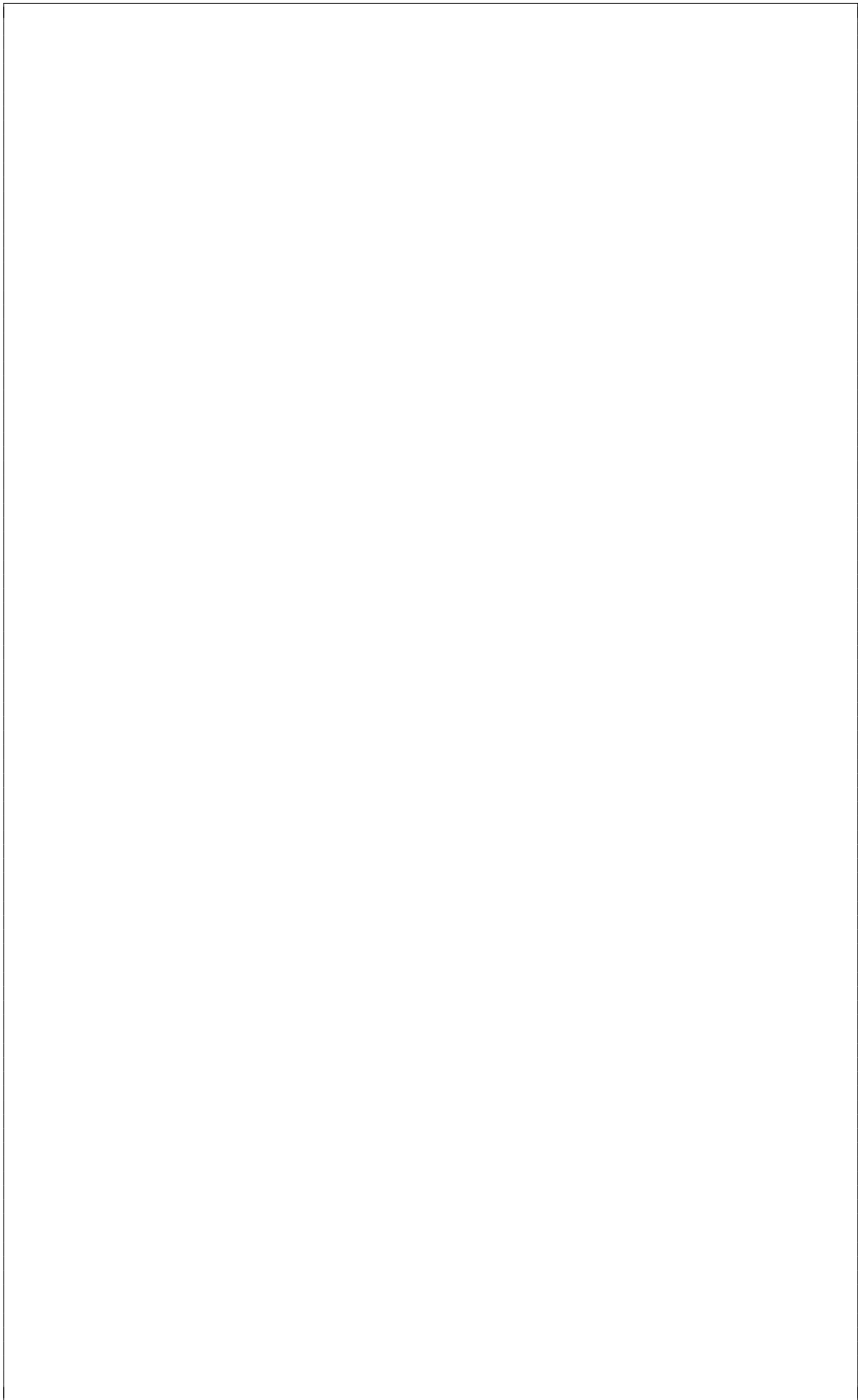
- (i) der Gesamtanzahl $X := \sum_{(i,j) \in [k]^2} |S_{i,j}|$ der Tupel in den Mengen; sowie
- (ii) der maximalen Anzahl $Y := \max_{(i,j) \in [k]^2} |S_{i,j}|$ an Tupeln in einer der Mengen?

Hinweis: Aus der Reduktion in der Vorlesung wissen wir, dass Clique selbst dann noch NP-schwer ist, wenn $k \in \Theta(|V|)$ gilt. Sie dürfen also annehmen, dass es eine (globale) Konstante $c > 0$ gibt, sodass $c \cdot n \leq k \leq n$.











CHRISTIAN-ALBRECHTS-UNIVERSITÄT ZU KIEL

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

14. Juli 2023

Modulprüfung zur Vorlesung »Analyse von Algorithmen und Komplexität (Komplexitätsteil)«

SS 2023

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, (bunte) Stifte (kein Rot, kein Grün, kein Bleistift).
- Einsichtnahme: Nach Absprache.

Aufgabe 1 RUCKSACKPROBLEM (3+2+5 Punkte)

Wir betrachten die Anwendung der Approximationsheuristiken für das Rucksackproblem aus der Vorlesung.

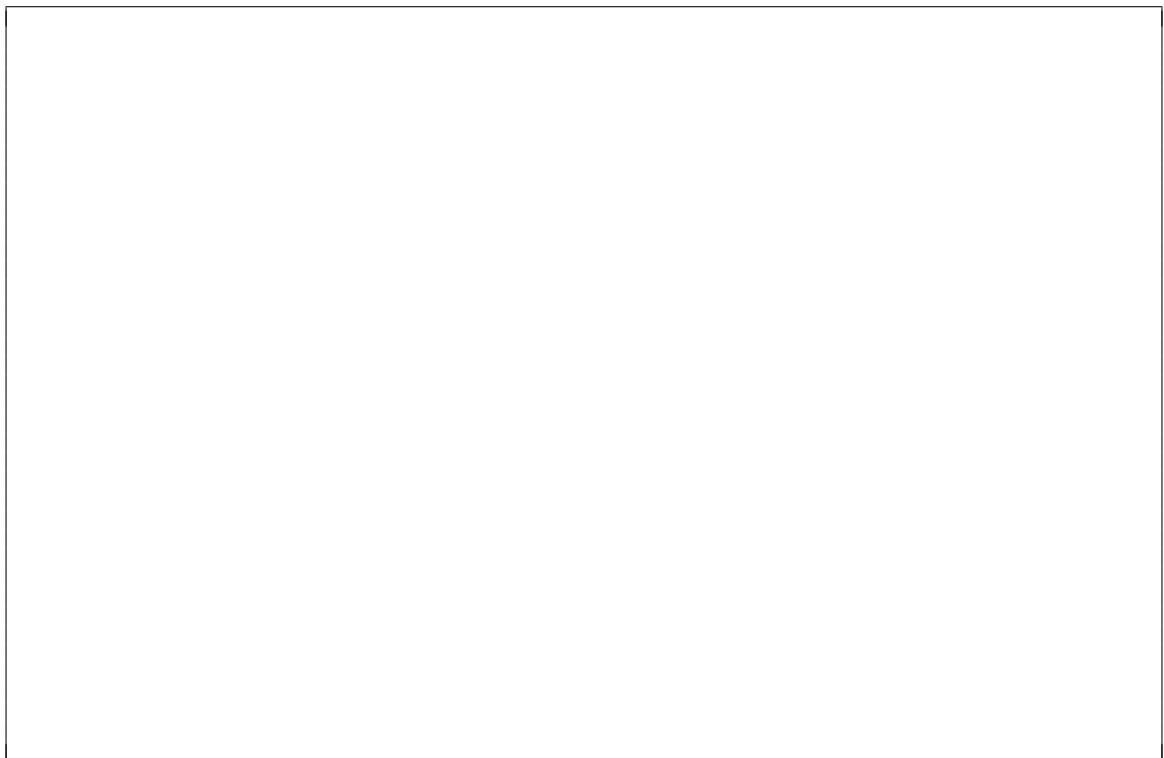
Gegeben sei dafür folgende Instanz des Rucksackproblems mit Kapazität $B = 16$. Items sind in der Form (p_i, w_i) angegeben.

$$I = [(1,4), (1,1), (1,3), (11,13), (3,3), (5,6), (1,5), 16]$$

- a) Wenden Sie den normalen Greedy-Algorithmus auf obige Instanz an. Begründen Sie kurz, warum die gewählten Items ausgewählt werden. Was ist der Lösungswert?



- b) Wenden Sie nun den ModifiedGreedy-Algorithmus an. Wie ändert sich die Lösung?



- c) Was ist die Güte des ModifiedGreedy-Algorithmus? Geben Sie eine kurze Idee an, wie man diesen Ansatz modifizieren kann, um die Güte auf $\frac{3}{2}$ oder $\frac{4}{3}$ zu verbessern.



Aufgabe 2 LONGEST PATH (3+3+4 Punkte)

Betrachten Sie das folgende Problem:

Problem: LONGEST PATH

Eingabe: Eine ungerichteter Graph $G = (V, E)$ und eine Zahl k .

Entscheide: Gibt es einen Pfad in G , der mindestens Länge k hat?

Zeigen Sie auf verschiedene Weisen, dass das Problem in NP liegt:

- Geben Sie ein Zertifikat für LONGEST PATH an.
- Beschreiben Sie einen Verifizierer für LONGEST PATH und geben Sie dessen Laufzeit konkret an.
- Beschreiben Sie eine nicht-deterministische Turing-Maschine (in Worten), die LONGEST PATH löst. Geben Sie auch hier die Laufzeit konkret an.



Aufgabe 3 APPROXIMATIVE ALGORITHMEN (6+4 Punkte)

Das Problem MAX-3-SAT sei folgendermaßen definiert:

Problem: MAX-3-SAT

Eingabe: Eine Formel ϕ in konjunktiver Normalform, wobei jede Klausel drei Literale enthält.

Ausgabe: Eine Belegung β der Variablen, die die Anzahl $v(\beta)$ der erfüllten Klauseln maximiert.

Betrachten Sie folgenden Algorithmus A:

- Sei β_0 die Belegung, die alle Variablen auf false setzt.
- Sei β_1 die Belegung, die alle Variablen auf true setzt.
- Falls $v(\beta_0) \geq v(\beta_1)$, gib β_0 zurück.
- Sonst gib β_1 zurück.

a) Zeigen Sie, dass obiger Algorithmus Güte 2 hat, d.h. $A(\phi) \geq \frac{1}{2}OPT(\phi)$ gilt für alle Eingaben ϕ .

b) Geben Sie eine Formel an, bei der der Algorithmus eine Belegung ausgibt, die genau die Hälfte der maximal erfüllbaren Klauseln erfüllt. Begründen Sie kurz, warum Ihre Formel geeignet ist.

Hinweis: Sie brauchen nur höchstens zwei Klauseln und sechs Variablen.



Aufgabe 4 BEWEIS ZUR VORLESUNG (10 Punkte)

Betrachten Sie folgendes Problem aus der Vorlesung:

Problem: k -CLIQUE

Eingabe: Ein ungerichteter Graph $G = (V, E)$ und eine Zahl $k \geq 1$. Eine Clique ist eine Teilmenge $C \subseteq V$ mit $\{u, v\} \in E$ für alle $u, v \in C$ mit $u \neq v$.

Entscheide: Hat der gegebene Graph G eine Clique $C \subseteq V$ mit mindestens k Knoten (d.h. mit $|C| \geq k$)?

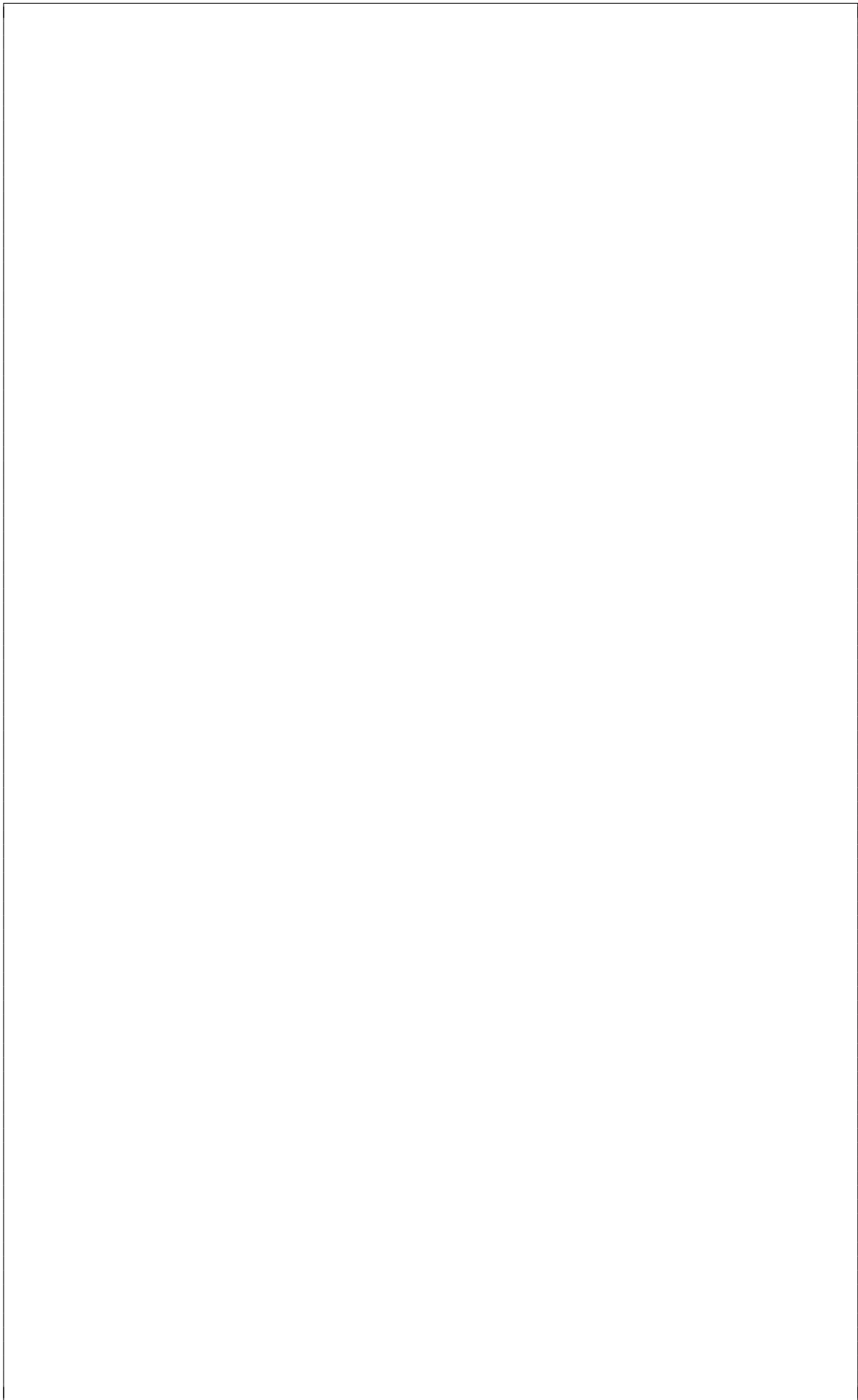
Beweisen Sie: Das Problem k -CLIQUE ist NP-vollständig.



Aufgabe 5 NP UND NP-VOLLSTÄNDIGKEIT (4+4+7 Punkte)

- a) Beweisen Sie die NP-Vollständigkeit von SUBSETSUM, bei der jede Itemgröße jeweils durch 3 oder durch 7 teilbar ist.
- b) Beweisen Sie die NP-Vollständigkeit von 3-COLOR, wobei jeder Knoten im Graph mindestens Grad 3 hat.
- c) Zeigen Sie: Für kein $\alpha > 1$ gibt es einen approximativen Algorithmus mit Güte α (d.h. mit Zielfunktionswert $\leq \alpha \cdot \text{OPT}$) für TSP, außer $P = NP$ (Denken Sie an Wiliam Rowan Hamilton).





Aufgabe 6 UNTERE SCHRANKEN UND ETH (7+8 Punkte)

Betrachten Sie folgende Reduktionen:

Problem: HITTINGSET

Eingabe: Eine Menge U sowie r Teilmengen $F_1, \dots, F_r \subseteq U$ und eine Zahl k .

Entscheide: Gibt es eine Menge $S \subseteq U$ mit $|S| \leq k$ und $S \cap F_i \neq \emptyset$ für alle $i \in [r]$?

- a) 3-SAT $\preceq$ HITTINGSET: Seien $v_1, \dots, v_n$ die Variablen der Formel und $C_1, \dots, C_m$ die Klauseln. Setze $U = \{v_1, \bar{v}_1, \dots, v_n, \bar{v}_n\}$ und $k = n$. Für jede Variable v_i definiere eine Menge $F_i = \{v_i, \bar{v}_i\}$ und für jede Klausel C_j definiere eine Menge $F_{n+j} = C_j$

Welche unteren Schranken ergeben sich durch obige Reduktion für die Laufzeit von Algorithmen für HITTINGSET unter Annahme der ETH in Abhängigkeit von

- (i) r ?
- (ii) $|U|$?

Beweisen Sie die entsprechenden unteren Schranken!

Hinweis: Sie dürfen – wie in den Hausaufgaben zur ETH – Faktoren, die polynomiell in der Eingabekodierung sind, vernachlässigen.

- b) 3-SAT $\preceq$ SUBSETSUM: Seien $v_1, \dots, v_n$ die Variablen der Formel und $C_1, \dots, C_m$ die Klauseln. Für jede Variable v_i erzeuge zwei Items a_i und b_i , mit den Größen

$$s(a_i) = 10^{i-1} + \sum_{\substack{j \in [m] \\ x_i \in C_j}} 10^{n+j-1} \quad \text{und} \quad s(b_i) = 10^{i-1} + \sum_{\substack{j \in [m] \\ \bar{x}_i \in C_j}} 10^{n+j-1}.$$

Zusätzlich erzeuge zwei Items c_j und d_j für jede Klausel C_j mit Größen

$$s(c_j) = s(d_j) = 10^{n+j-1}.$$

Der Zielwert sei

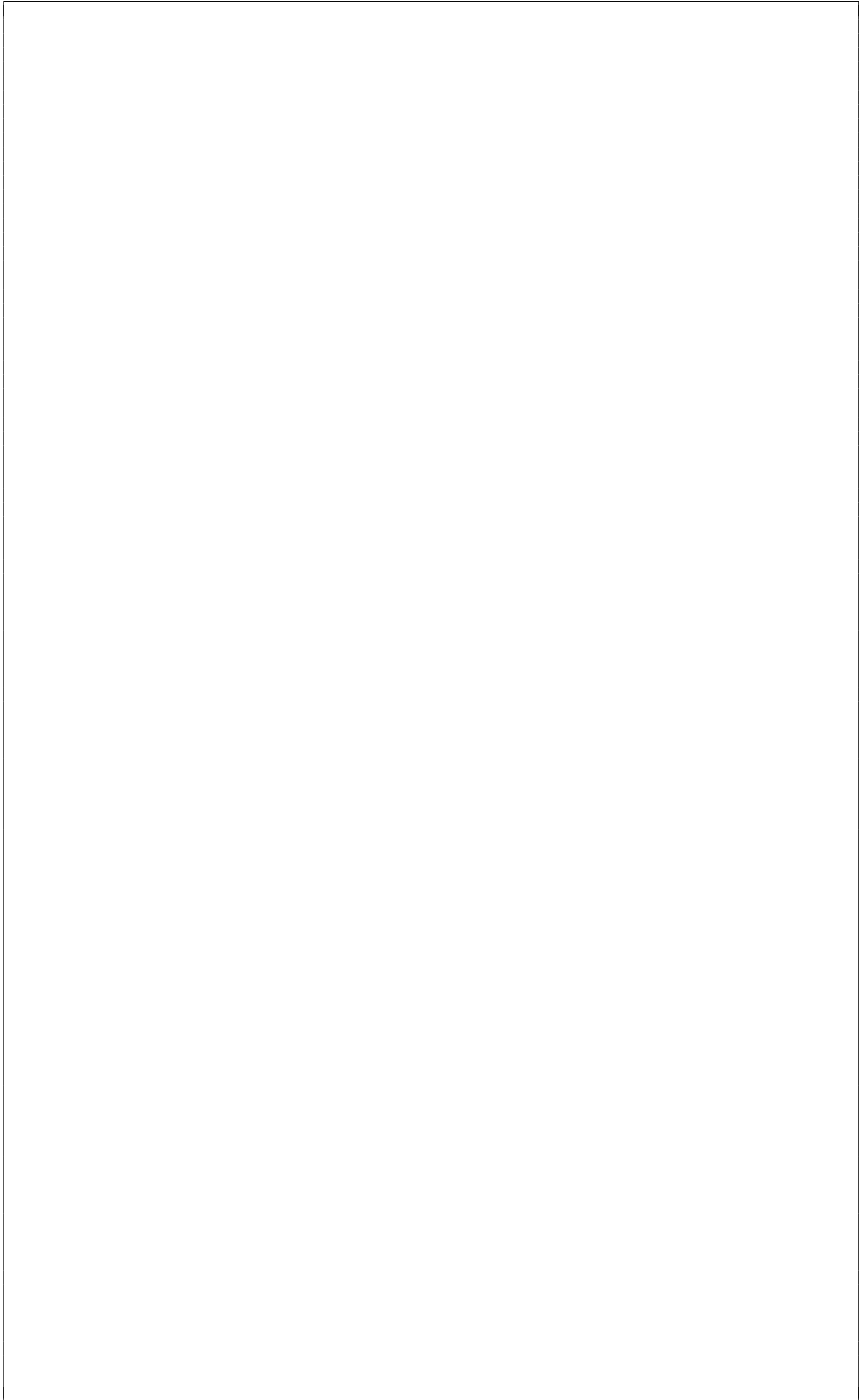
$$B = \sum_{j \in [m]} (3 \cdot 10^{n+j-1}) + \sum_{i \in [n]} 10^{i-1}.$$

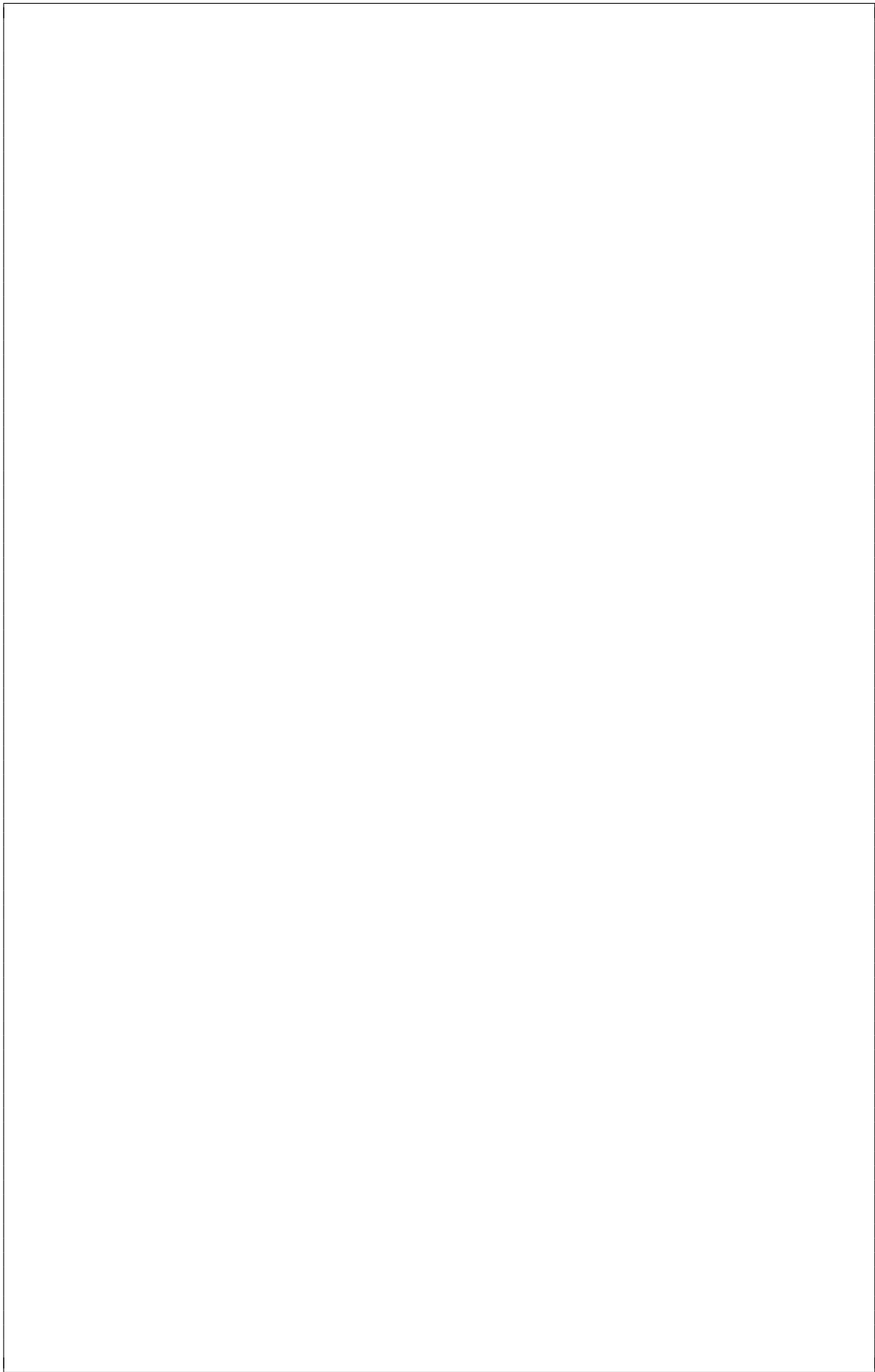
Welche unteren Schranken ergeben sich durch obige Reduktion für die Laufzeit von Algorithmen für SUBSETSUM unter Annahme der ETH in Abhängigkeit von

- (i) der Anzahl der unterschiedlichen Itemgrößen d ?
- (ii) der binären Kodierungslänge L der größten Itemgröße Δ (d.h. $L = \log(\Delta)$)?

Beweisen Sie die entsprechenden unteren Schranken!

Hinweis: Sie dürfen auch hier – wie in den Hausaufgaben zur ETH – Faktoren, die polynomiell in der Eingabekodierung sind, vernachlässigen.





Viel Erfolg!



CHRISTIAN-ALBRECHTS-UNIVERSITÄT ZU KIEL

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

19. Oktober 2023

Modulprüfung zur Vorlesung »Analyse von Algorithmen und Komplexität (Komplexitätsteil)«

SS 2023

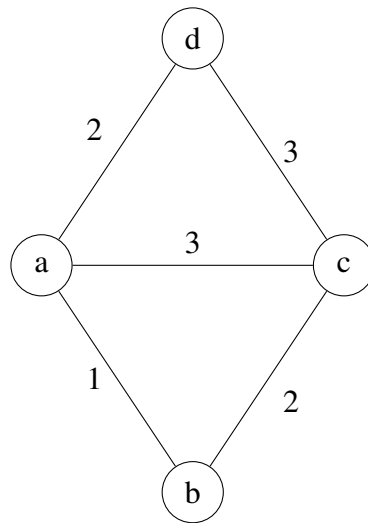
Name: _____ Matrikel-Nr.: _____

Hinweise:

- Bearbeiten Sie *alle* Aufgaben.
- Sie haben 120 Minuten Zeit für die Klausurbearbeitung.
- Erlaubte Hilfsmittel: Ein einseitig handschriftlich beschriebenes Blatt Papier, (bunte) Stifte (kein Rot, kein Grün, kein Bleistift, kein Tintenkiller, kein Tipp-Ex).
- Schreiben Sie auf jedes Blatt ihren Namen.
- Wenn Sie Schmierzettel verwenden um eine Aufgabe zu bearbeiten, verweisen Sie in der Aufgabe auf diese.
- Einsichtnahme: 2.11.23, 13:30-14:30 Uhr.
- Die Klausur hat 6 Aufgaben auf 13 Seiten – überprüfen Sie bitte vor der Bearbeitung, dass Ihre Klausur vollständig ist.

Aufgabe 1 CHRISTOFIDES' ALGORITHMUS (3+7 Punkte)

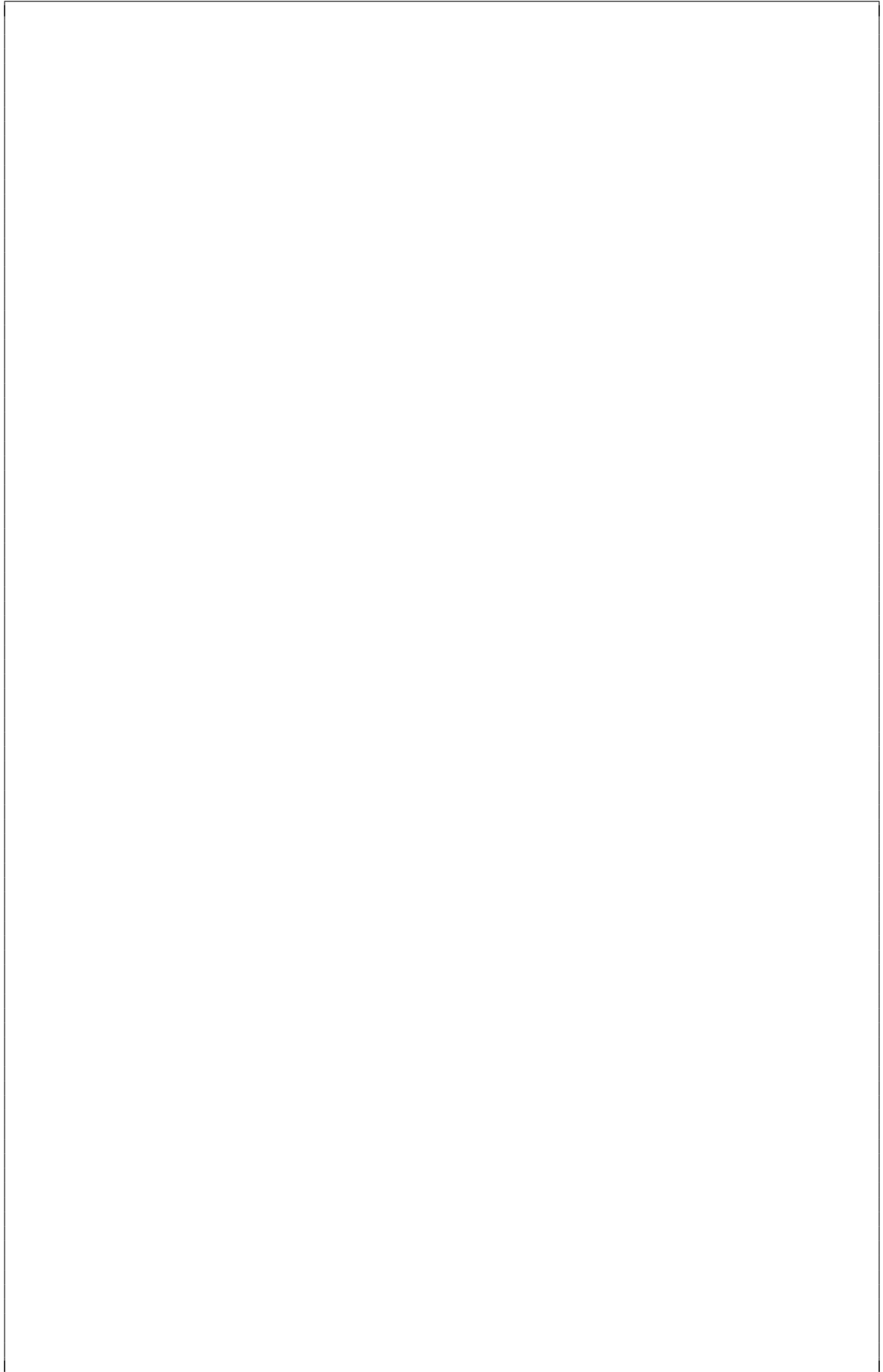
Gegeben sei folgender Graph $G = (V, E)$:



- a) Geben Sie an, was Christofides' Algorithmus berechnet, und welche approximative Güte er hat. Welche Eigenschaft müssen die Distanzen neben der Symmetrie noch erfüllen, damit Christofides' Algorithmus korrekt arbeitet.



- b) Wenden Sie den Algorithmus von Christofides 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.



Aufgabe 2 DOMINATING SET (3+3+4 Punkte)

Betrachten Sie das folgende Problem:

Problem: DOMINATING SET

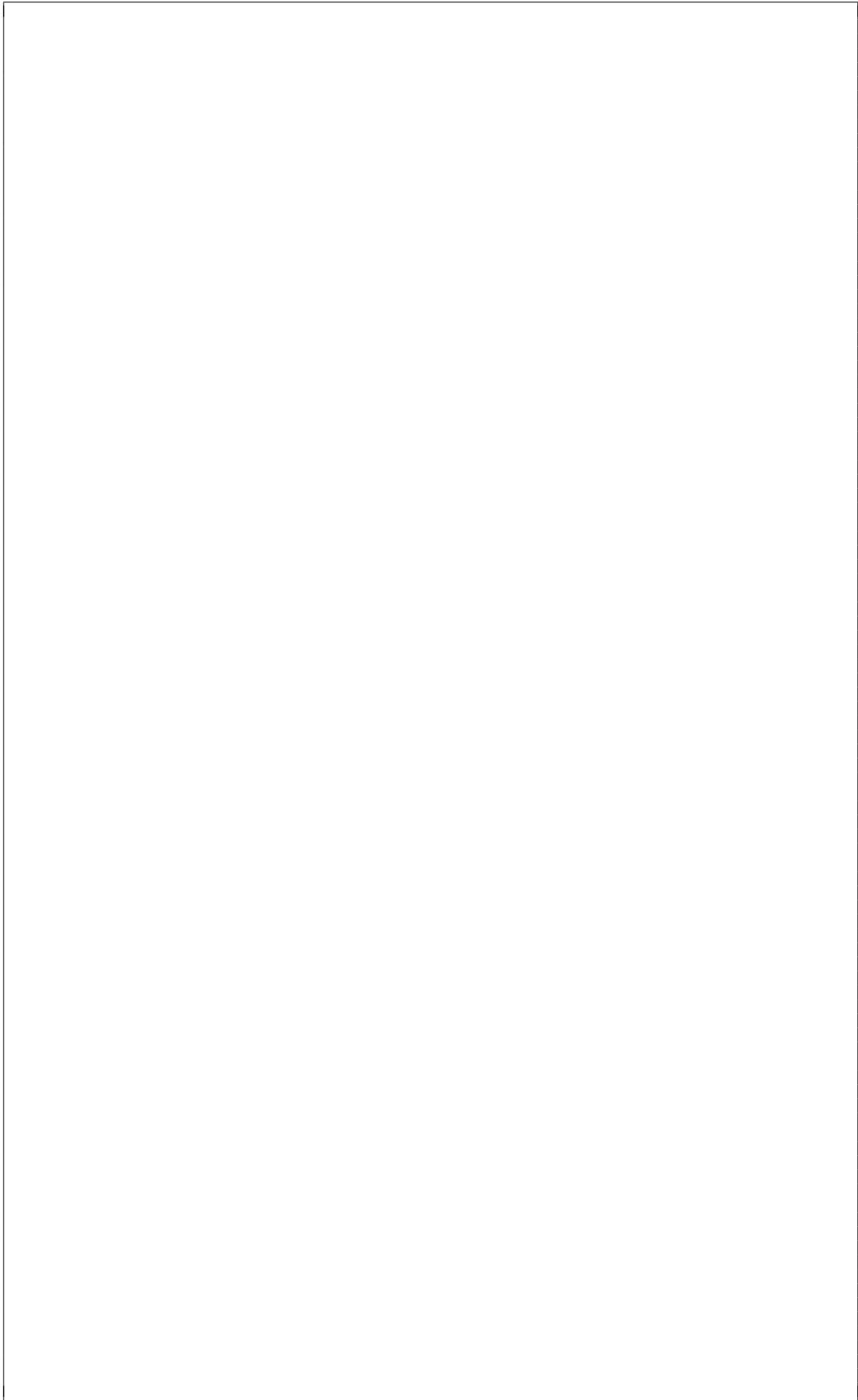
Eingabe: Ein ungerichteter Graph $G = (V, E)$ (wobei $V = \{1, \dots, n\}$ für eine Zahl $n \in \mathbb{N}_{\geq 1}$) und eine Zahl $k \in \mathbb{N}_0$.

Entscheide: Gibt es eine Teilmenge $D \subseteq V$ der Knoten mit $|D| \leq k$, sodass jeder Knoten $v \in V$ in D enthalten ist oder benachbart zu mindestens einem der Knoten in D ist?

Zeigen Sie auf verschiedene Weisen, dass das Problem in NP liegt:

- Geben Sie ein Zertifikat für DOMINATING SET an.
- Beschreiben Sie einen polynomiellen Verifizierer für DOMINATING SET und schätzen Sie die Laufzeit in $\mathcal{O}$ -Notation konkret ab.
- Beschreiben Sie (in Worten) eine polynomielle, nicht-deterministische Turing-Maschine, die DOMINATING SET löst. Schätzen Sie auch hier die Laufzeit in $\mathcal{O}$ -Notation konkret ab.





Aufgabe 3 APPROXIMATIVE ALGORITHMEN (6+4 Punkte)

Das Problem MIN-EDGE-COVER sei folgendermaßen definiert:

Problem: MIN-EDGE-COVER

Eingabe: Ein zusammenhängender, ungerichteter Graph $G = (V, E)$.

Ausgabe: Eine Teilmenge der Kanten $C \subseteq E$ mit minimaler Kardinalität, sodass jeder Knoten $v \in V$ zu mindestens einer Kante aus C inzident ist.

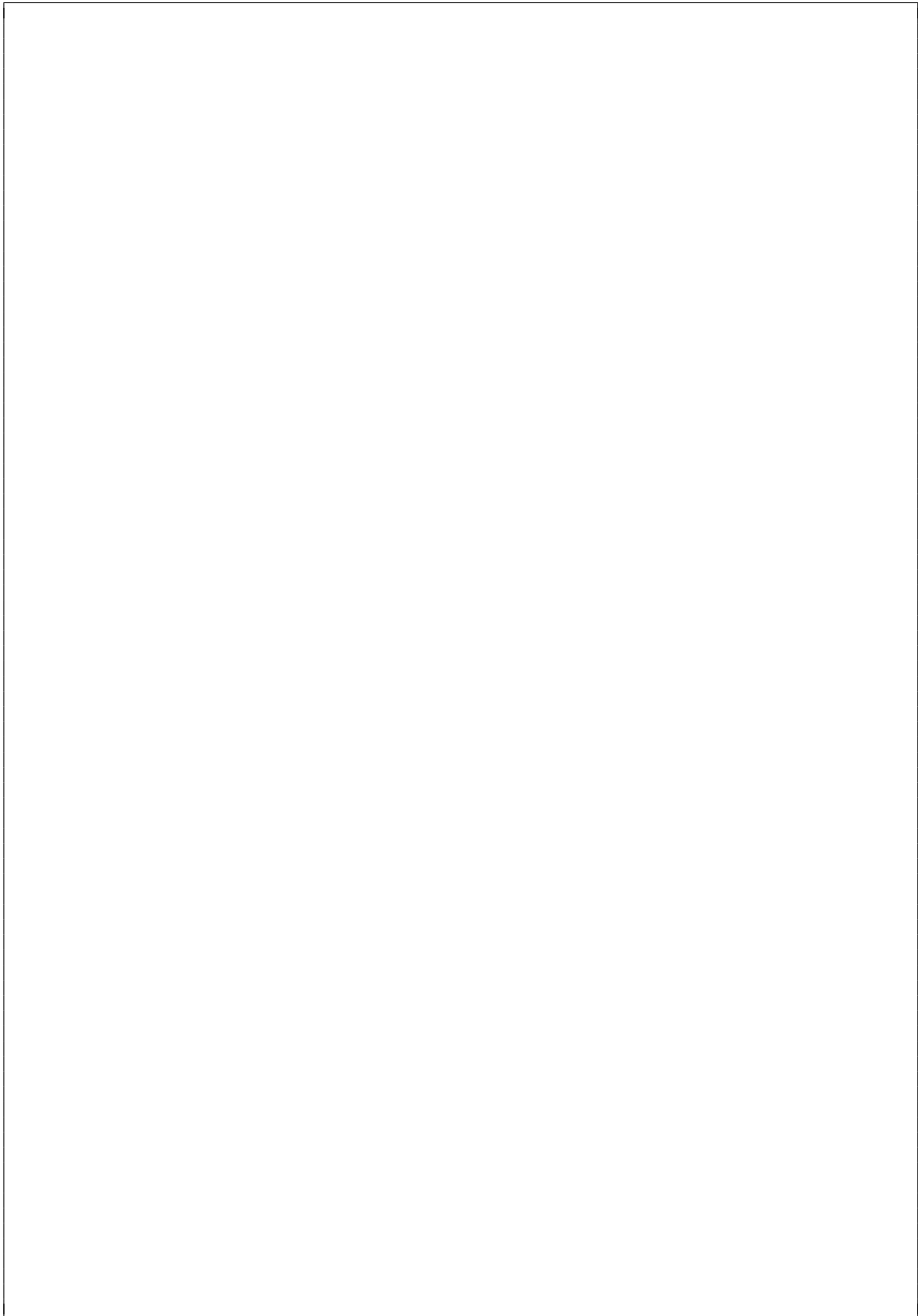
Betrachten Sie folgenden Algorithmus A:

- Gehe (in beliebiger Reihenfolge) alle Knoten durch:
- Falls der aktuelle Knoten v noch nicht abgedeckt ist, wähle zufällig eine der zu v inzidenten Kanten und füge sie zu C hinzu.
- Gib anschließend die entstehende Menge C zurück.

a) Zeigen Sie, dass obiger Algorithmus Güte 2 hat, d.h. $A(G) \leq 2 \cdot \text{OPT}(G)$ gilt für alle Eingaben G .



b) Sei $n \in \mathbb{N}_{\geq 4}$ gerade. Betrachten Sie den Graphen G_n , der genau n Knoten und n Kanten hat und einen Kreis beschreibt. D.h. $G_n = (\{1, \dots, n\}, \{\{i, i+1\} \mid i \in \{1, \dots, n-1\}\} \cup \{\{n, 1\}\})$. Wie viele Kanten hat ein MIN-EDGE-COVER des Graphen G_n ? Wie schlecht ist die Approximationsrate des Algorithmus A für G_n im schlimmsten Fall? Geben Sie eine Reihenfolge der Knoten (für Schritt 1) und eine Auswahl der Kanten (für Schritt 2) an, die zu einer möglichst schlechten Approximationsrate führen.




Aufgabe 4 BEWEIS – SCHEDULING AUF IDENTISCHEN MASCHINEN (10 Punkte)

Betrachten Sie das Scheduling-Problem $P||C_{\max}$:

Problem: $P||C_{\max}$

Eingabe: Eine Liste $L = (J_1, \dots, J_n)$ von n Jobs mit Ausführungszeiten $p_1, \dots, p_n \in \mathbb{N}$, und m identische Maschinen.

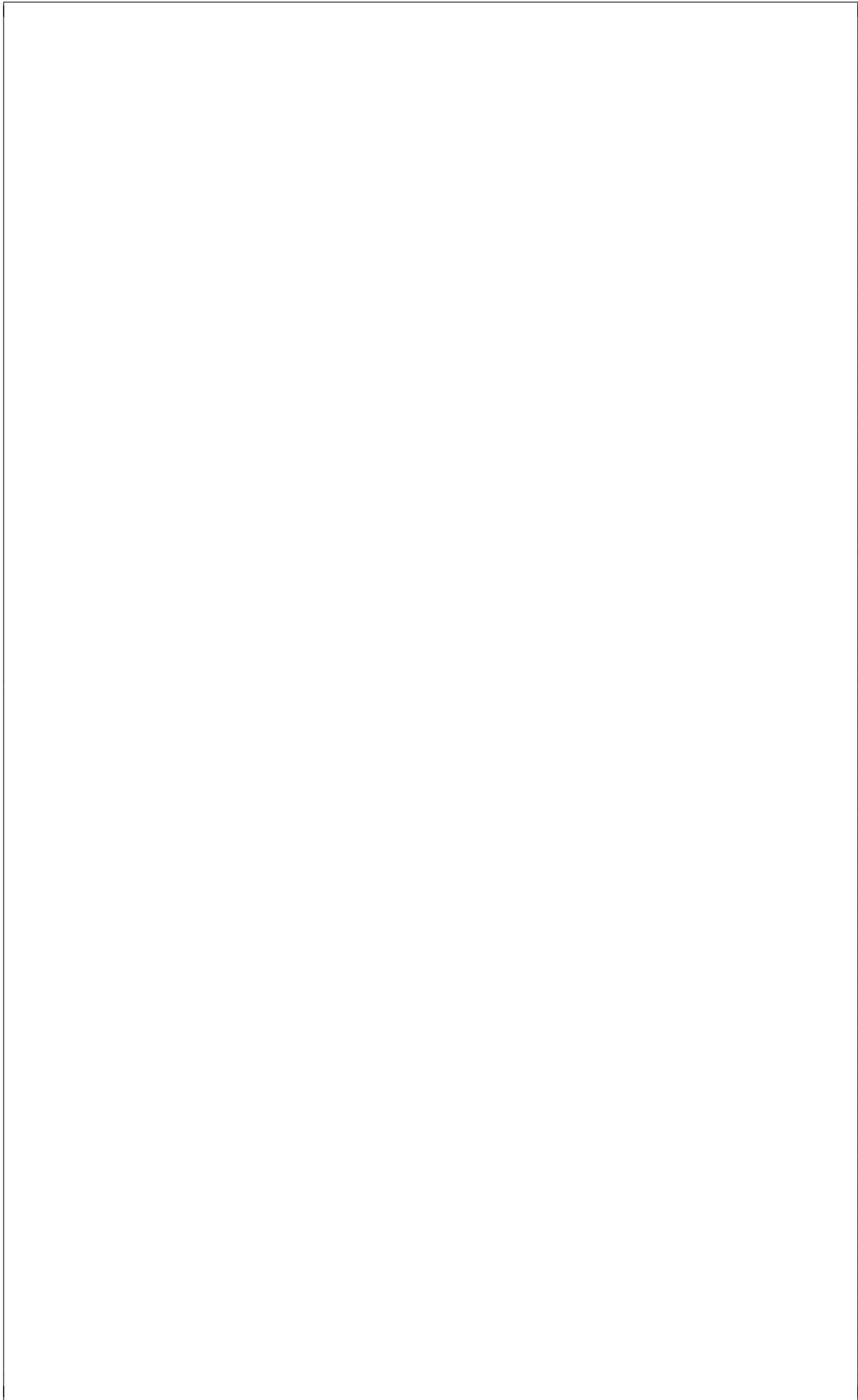
Ausgabe: Ein Schedule (Partition von J in m Teilmengen $B_1, \dots, B_m$) mit **minimaler maximaler**

Last $C_{\max} := \max_{1 \leq i \leq m} \sum_{J_j \in B_i} p_j$.

Betrachten Sie den folgenden Algorithmus ListScheduling, der einen Schedule berechnet. Sei $LS(I)$ die Last $C_{\max}$ eines Schedules des Algorithmus, und $OPT(I)$ die Last $C_{\max}$ eines optimalen Schedules zur Eingabe I . Zeigen Sie: Für alle Eingaben $I = (L, m)$ gilt $LS(I)/OPT(I) \leq 2 - 1/m$.

Algorithmus LISTSCHEDULING ($L = (J_1, \dots, J_n), m$)

```
1  for  $i = 1$  to  $m$  do
2     $E_i = 0$ ;  $B_i = \emptyset$ ;
3  od
4  for  $j = 1$  to  $n$  do
5    wähle Job  $J_j$  aus Liste  $L$ ;
6    wähle Maschine  $M_i$  mit minimaler Last  $E_i$ ;
7     $B_i = B_i \cup \{J_j\}$ ;
8     $E_i = E_i + p_j$ ;
9  od
```



Aufgabe 5 NP UND NP-VOLLSTÄNDIGKEIT (4+5+6 Punkte)

Problem: HAMILTONIANPATH

Eingabe: Ein ungerichteter Graph $G = (V, E)$

Entscheide: Existiert ein Pfad in G , der jeden Knoten genau einmal besucht?

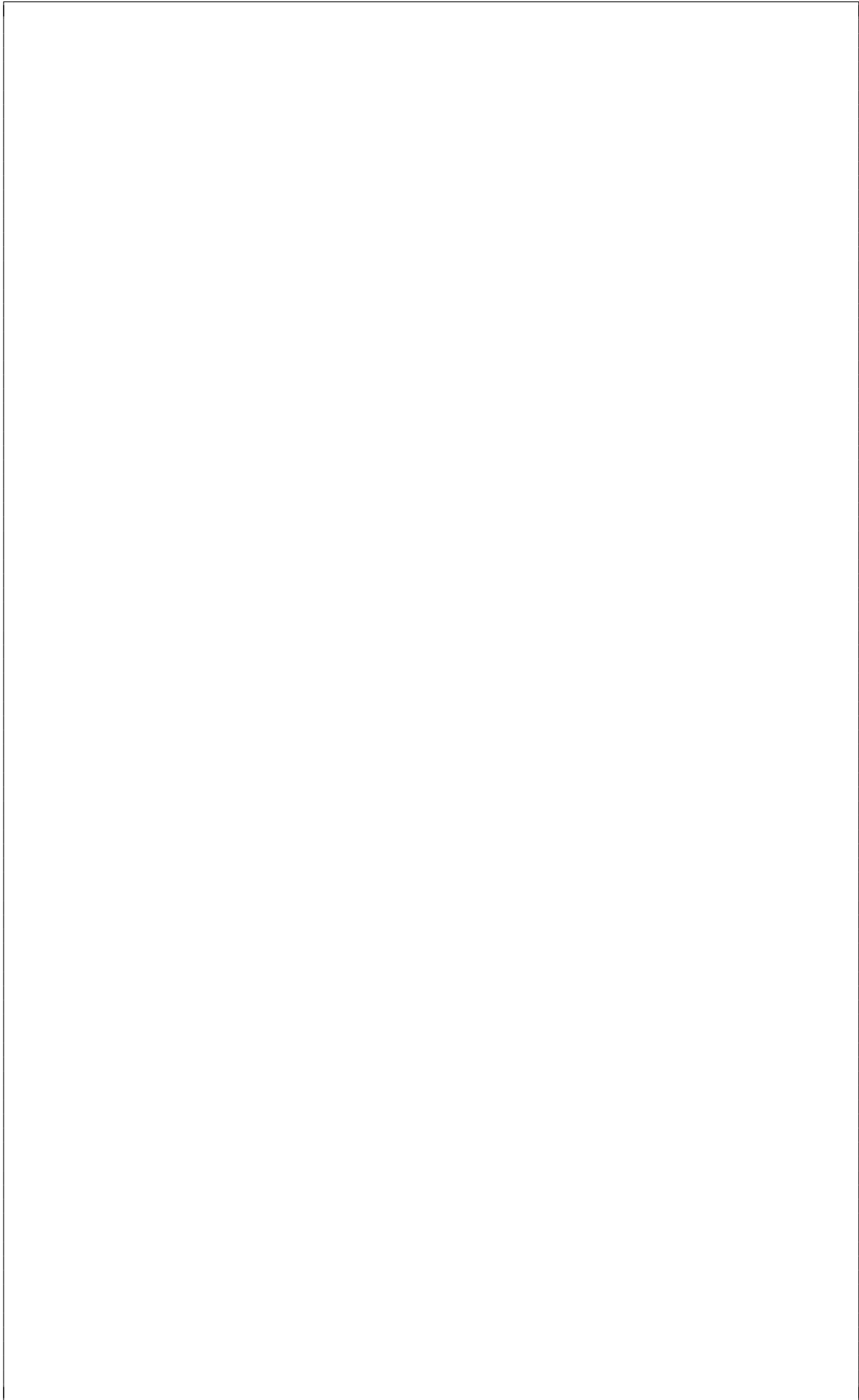
Problem: HAMILTONIANCYCLE

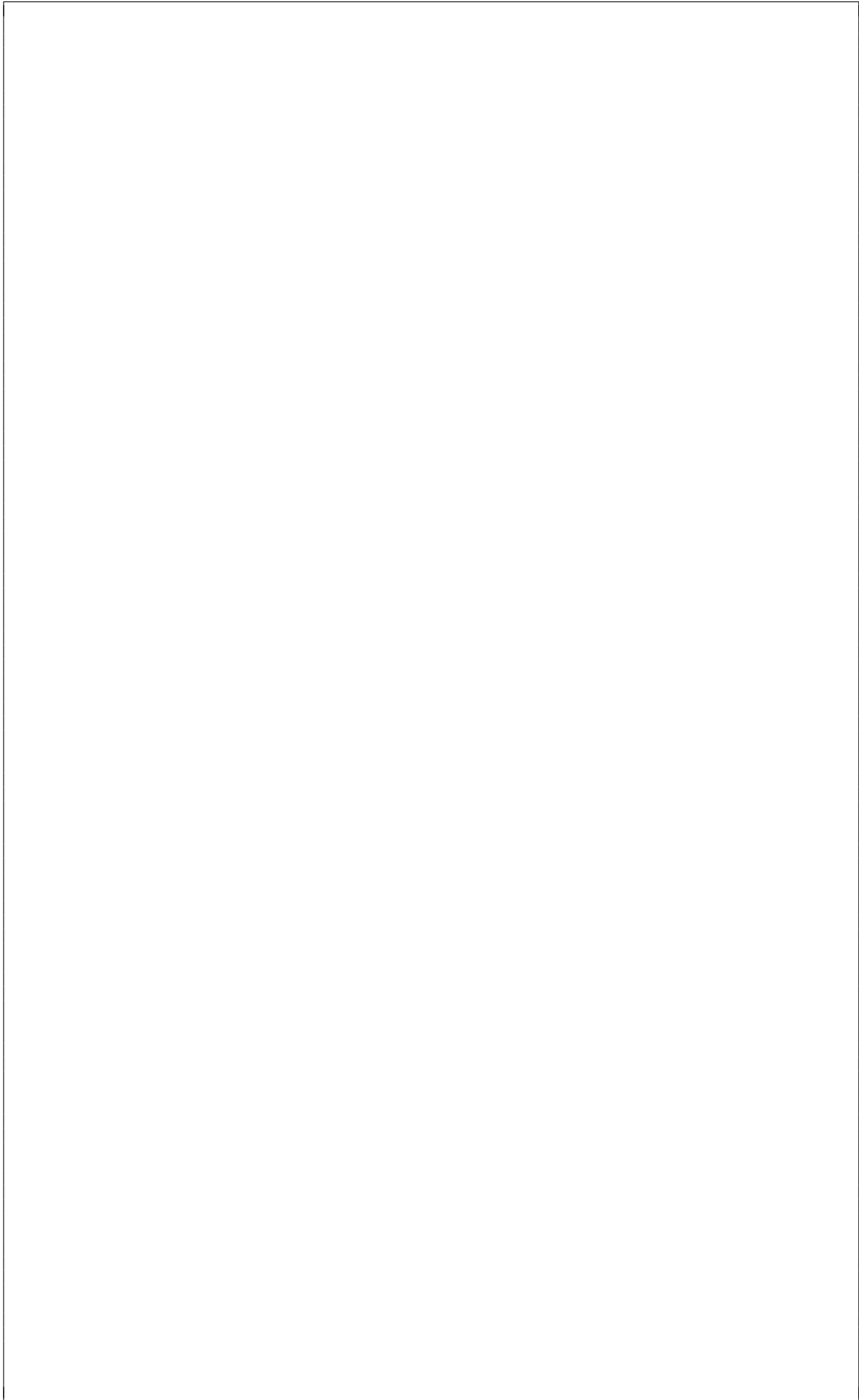
Eingabe: Ein ungerichteter Graph $G = (V, E)$

Entscheide: Existiert ein Kreis in G , der jeden Knoten genau einmal besucht?

- (a) Sei $\mathcal{L} \subseteq \text{SUBSETSUM}$ die Menge der (positiven) SubsetSum-Instanzen, bei denen *keine* der Itemgrößen eine Zweierpotenz ist. Zeigen Sie, dass $\mathcal{L}$ NP-vollständig ist.
- (b) Geben Sie eine polynomielle Reduktion von HAMILTONIANPATH auf HAMILTONIANCYCLE an.
- (c) Geben Sie eine polynomielle Reduktion von HAMILTONIANCYCLE auf HAMILTONIANPATH an.







Aufgabe 6 UNTERE SCHRANKEN UND ETH (5+5 Punkte)

Im Folgenden bezeichne $\langle I \rangle$ die Kodierungslänge der Instanz.

(a) Wir betrachten folgendes Problem:

Problem: HITTINGSET

Eingabe: Eine Menge U' sowie r' Teilmengen $F'_1, \dots, F'_{r'} \subseteq U'$ und eine Zahl k' .

Entscheide: Gibt es eine Menge $S' \subseteq U'$ mit $|S'| \leq k'$ und $S' \cap F'_i \neq \emptyset$ für alle $i' \in \{1, \dots, r'\}$?

Sie dürfen nutzen, dass sich HITTINGSET unter Annahme der ETH nicht in Zeit $2^{o(r')} \langle I \rangle^{\mathcal{O}(1)}$ und nicht in Zeit $2^{o(|U'|)} \langle I \rangle^{\mathcal{O}(1)}$ lösen lässt.

Problem: SETCOVER

Eingabe: Eine Menge U sowie r Teilmengen $F_1, \dots, F_r \subseteq U$ und eine Zahl k .

Entscheide: Gibt es eine Menge $S \subseteq \{1, \dots, r\}$ mit $|S| \leq k$ und $\bigcup_{i \in S} F_i = U$?

Welche unteren Schranken ergeben sich

- (i) in Hinblick auf $|U|$ sowie
- (ii) in Hinblick auf r

für SETCOVER unter Annahme der ETH aus den unteren Schranken für HITTINGSET, wenn man folgende Reduktion von SetCover auf HittingSet verwendet?

Sei $(\mathcal{F}, U, k)$ eine SETCOVER-Instanz. Setze $V_1 := U$ und $V_2 := \{1, \dots, r\}$. Sei $V := V_1 \dot{\cup} V_2$. Weiter sei $E = \{\{v, w\} \mid v \in V_1, w \in V_2, v \in F_w\}$. Damit ist $G = (V, E)$ ein bipartiter Graph mit Partitionen V_1, V_2 . Setze $U' := \{1, \dots, r\}$. Erzeuge nun eine Menge F'_w für jeden Knoten $w \in V_1$, indem $F'_w := \{v \in V_2 \mid \{v, w\} \in E\}$ gesetzt wird. Sei $\mathcal{F}' := \{F'_w \mid w \in V_1\}$. Mit $k' = k$ ist die berechnete Instanz nun durch $(\mathcal{F}', U', k')$ gegeben.



- (b) Wir wollen uns mit einem Problem beschäftigen, das ein mächtiges Werkzeug im Bereich der Optimierung ist: Integrale Lineare Programme (ILPs).

Problem: ILP-FEASIBILITY

Eingabe: Eine Matrix $A \in \mathbb{Z}^{M \times N}$, eine rechte Seite $b \in \mathbb{Z}^M$.

Entscheide: Gibt es einen Vektor $x \in \mathbb{Z}_{\geq 0}^N$ mit $Ax \leq b$, d.h. $A_i x_i \leq b_i$ für alle $i \in \{1, \dots, M\}$?

Welche unteren Schranken ergeben sich

- (i) in Hinblick auf M sowie
- (ii) im Hinblick auf N

für ILP-FEASIBILITY unter Annahme der ETH, wenn man folgende Reduktion von 3-SAT auf ILP-FEASIBILITY verwendet?

Sei ϕ eine 3-SAT Instanz mit m Klauseln $(C_i)_{i \in m}$ und n Variablen. Analog zu linearen Gleichungssystemen ($Ax = b$), die aus M Gleichungen bestehen, haben wir bei $Ax \leq b$ einen Satz von M linearen Ungleichungen. Wir konstruieren diese wie folgt: Wir nutzen $N := 2n$ Variablen in unserem ILP, eine für jedes (mögliche) Literal in der SAT-Formel. Für jede SAT-Variable v sei x^v die ILP-Variable, die zum SAT-Literal v korrespondiert, und $x^{\bar{v}}$ die, die zum SAT-Literal $\bar{v}$ korrespondiert. Für jede Klausel C_i erzeugen wir eine Ungleichung $-\sum_{\ell \in C_i} x^\ell \leq -1$. Weiter erzeugen wir für jede SAT-Variable v zwei Ungleichungen: $x^v + x^{\bar{v}} \leq 1$ und $-x^v - x^{\bar{v}} \leq -1$. Durch das Eintragen der entsprechenden Koeffizienten in eine erweiterte Koeffizientenmatrix (bestehend aus Matrix A und rechter Seite b) erhalten wir die ILP-FEASIBILITY Instanz.

Viel Erfolg!