Algorithmen und Datenstrukturen 2
WiSe 2023/24 · Prof. Dr. Dell
13. Februar 2024
Klausurnummer: **0000**

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Nachname (Druckschrift): ______________________

Vorname (Druckschrift): ______________________

Matrikelnummer: ______________________

Studiengang: ______________________

Bitte Hinweise beachten:

- Schreiben Sie Ihren Namen nur auf dieses Titelblatt.
- Zusatzpapier darf nicht mit abgegeben werden.
- Zugelassenes Hilfsmittel: 1 Blatt DIN A4.
- Nicht zugelassene Hilfsmittel (z. B. Handys, Smartwatches, andere Geräte) stellen eine Täuschung dar und führen zum Nichtbestehen der Klausur. Schalten Sie daher alle elektronischen Geräte aus und verstauen Sie diese in Ihrer Tasche.
- Werden zu einer Aufgabe zwei oder mehr Lösungen angegeben, so gilt die Aufgabe als nicht gelöst. Entscheiden Sie sich also immer für **eine** Lösung.
- Begründungen sind nur dann notwendig, wenn die Aufgabenformulierung dies verlangt.
- Die Klausur dauert 180 Minuten und gilt mit 50 % der Höchstpunktzahl als bestanden.
- In allen Multiple-Choice-Fragen sind **genau zwei von fünf Antworten** richtig. Wenn man x richtige Kreuze setzt und y falsche, erhält man entweder 0, 1 oder 2 Punkte:

richtige Kreuze *(correct marks)*	x	**2**	**2**	2	2	**1**	1	1	1	0	0	0	0
falsche Kreuze *(wrong marks)*	y	**0**	**1**	2	3	**0**	1	2	3	0	1	2	3
Punkte *(points)*		**2**	**1**	0	0	**1**	0	0	0	0	0	0	0

Please note:

- *Write your name on this cover sheet only.*
- *Additional paper must not be submitted.*
- *Approved aid: 1 sheet DIN A4.*
- *Non-approved aids (e.g., mobile phones, smartwatches, other devices) constitute deception and lead to failing the exam. Turn your electronic devices off before the exam and store them in your bag.*
- *If two or more solutions are given for a problem, the problem counts as unsolved. So always choose **one** solution.*
- *Justifications are only necessary if the problem wording requires it.*
- *The exam lasts 180 minutes and is considered passed with 50 % of the maximum score.*
- *In all multiple-choice questions, **exactly two out of five answers** are correct. If you make x correct and y wrong marks, you will receive 0, 1, or 2 points according to the table above.*

Diese Seite ist für den internen Gebrauch.
Bitte leer lassen.

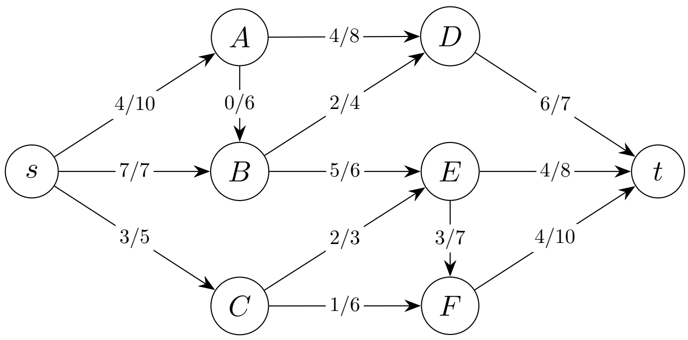
Diese Seite ist für den internen Gebrauch.
Bitte leer lassen.

Aufgabe	1	2	3	4	5
Erreichbar	20	20	20	20	20
Erreicht					

Summe	Note
/100	

Betrachten Sie den s-t-Fluss f in dem folgenden Flussnetzwerk (G, s, t, c):

Consider the s-t-flow f in the following flow network (G, s, t, c):



Jede Kante uv ist hierbei mit $f(uv)/c(uv)$ beschriftet, zum Beispiel trägt die Kante AD den Fluss $f(AD) = 4$ und hat die Kapazität $c(AD) = 8$.

Each edge uv is labeled with $f(uv)/c(uv)$, for example the edge AD carries the flow $f(AD) = 4$ and has capacity $c(AD) = 8$.

a) Welchen Wert hat ein **maximaler** Fluss in (G, s, t, c)?

*What is the value of a **maximum** flow in (G, s, t, c)?*

Antwort: 18.

(/ 2 Punkte)

b) Geben Sie die beiden Mengen S, T eines **minimalen** Schnittes in (G, s, t, c) an.

*Provide the two sets S, T of a **minimum** cut in (G, s, t, c).*

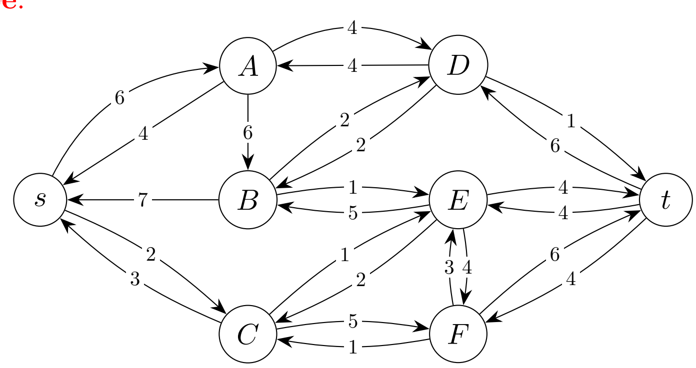
$S = \{s,$ A, B, D $\}$, $T = \{t,$ C, E, F $\}$.

(/ 2 Punkte)

c) Zeichnen Sie in der folgenden Abbildung den Residualgraphen G_f ein, der zum Flussnetzwerk (G, s, t, c) und dem Fluss f gehört. Geben Sie dabei die Restkapazitäten durch Beschriftung an jeder Kante von G_f an. Kanten mit Restkapazität 0 müssen **nicht** gezeichnet werden.

*In the following figure, draw the residual graph G_f for the flow network (G, s, t, c) and the flow f. Provide the residual capacities by labeling each edge of G_f. Edges with residual capacity 0 **don't** need to be drawn.*

Lösungsskizze:



(/ 3 Punkte)

d) Sei $G = (V, E)$ ein gerichteter Graph mit Kantengewichten, die positiv, null oder negativ sein können. Negative Kreise dürfen jedoch nicht vorkommen. Wir möchten den folgenden Wert Z berechnen: *Let $G = (V, E)$ be a directed graph with edge weights that can be positive, zero, or negative. However, negative cycles are not allowed to occur. We want to compute the following value Z:*

$$Z = \sum_{s \in V} \sum_{t \in V} \operatorname{dist}(s, t).$$

Dabei ist $\operatorname{dist}(s, t)$ die Länge eines kürzesten Weges von s nach t in G. Beschreiben Sie einen Algorithmus, der Z in $O(|V|^3)$ Zeit berechnet. Sie dürfen Algorithmen aus der Vorlesung verwenden, ohne deren Pseudocode anzugeben. Die Laufzeit muss nicht begründet werden.

Here, $\operatorname{dist}(s, t)$ is the length of a shortest path from s to t in G. Describe an algorithm that computes Z in $O(|V|^3)$ time. You may use algorithms from the lecture without providing their pseudocode. The running time doesn't need to be justified.

Lösungsskizze: Es handelt sich um eine direkte Anwendung des Floyd-Warshall-Algorithmus: Wir berechnen Z in $O(\|V\|^3)$ Zeit, indem wir den Floyd-Warshall-Algorithmus verwenden, um alle kürzesten Wege zwischen allen Knotenpaaren zu berechnen. Dann summieren wir die Längen aller kürzesten Wege auf, um Z zu erhalten.

(/ 3 Punkte)

e) Sei $B = (L, R, E)$ ein bipartiter Graph mit linker Seite L, rechter Seite R und der Kantenmenge $E \subseteq L \times R$. Eine Kantenmenge $S \subseteq E$ heißt *gut*, wenn jeder Knoten in L mit genau zwei Kanten in S inzident ist und jeder Knoten in R mit genau drei Kanten in S inzident ist.

Entwerfen Sie in den folgenden drei Schritten einen Algorithmus, der mithilfe eines maximalen Flusses entscheidet, ob ein gegebener bipartiter Graph B eine gute Kantenmenge enthält.

Let $B = (L, R, E)$ be a bipartite graph with left side L, right side R and edge set $E \subseteq L \times R$. A set of edges $S \subseteq E$ is called good *if every node in L is incident to exactly two edges in S and every node in R is incident to exactly three edges in S.*

In the following three steps, design an algorithm that, using maximum flow, decides whether a given bipartite graph B contains a good set of edges.

1) Beschreiben Sie ein geeignetes Flussnetzwerk $G = (V, E', s, t)$.
Describe a suitable flow network $G = (V, E', s, t)$.

Lösungsskizze: (Es handelt sich um eine Variante von „maximales bipartites Matching" aus den Folien (Folie 46ff „Network Flow II" von ALGO2 2023/24).)

(a) **Definition der Knoten**: Das Flussnetzwerk G besteht aus einer Menge von Knoten V, die alle Knoten aus L und R des bipartiten Graphen B, sowie zwei zusätzliche Knoten: eine Quelle s und eine Senke t, beinhaltet.

(b) **Definition der Kanten und Kapazitäten**: Die Kantenmenge E' des Flussnetzwerks wird wie folgt konstruiert:
- Von der Quelle s zu jedem Knoten in L führen Kanten mit einer Kapazität von 2, da jeder Knoten in L mit genau zwei Kanten in der guten Kantenmenge inzident sein muss.
- Von jedem Knoten in R führen Kanten zur Senke t mit einer Kapazität von 3, entsprechend der Anforderung, dass jeder Knoten in R mit genau drei Kanten in der guten Kantenmenge inzident sein muss.
- Zwischen den Knoten in L und R werden Kanten entsprechend der Kantenmenge E des bipartiten Graphen B eingerichtet, wobei jede dieser Kanten eine Kapazität von 1 hat, da jede Kante in der guten Kantenmenge nur einmal gezählt werden soll.

2) Beschreiben Sie in 1-2 Sätzen, wie Ihr Algorithmus nun mithilfe von G entscheidet, ob B eine gute Kantenmenge enthält. (Welchen Algorithmus aus der Vorlesung benutzen Sie und was macht Ihr Algorithmus mit dessen Ausgabe?)
Describe in 1-2 sentences how your algorithm now uses G to decide whether B contains a good set of edges. (Which algorithm from the lecture do you use and what does your algorithm do with its output?)

Lösungsskizze: Um zu entscheiden, ob B eine gute Kantenmenge enthält, nutzen wir den Ford-Fulkerson-Algorithmus zur Bestimmung des maximalen Flusses in G. Wenn der maximale Fluss in G sowohl der Summe der Kapazitäten der Kanten, die von der Quelle s ausgehen, als auch der Summe der Kapazitäten der Kanten, die in die Senke t münden, entspricht, dann enthält B eine gute Kantenmenge; andernfalls nicht.

3) Geben Sie die Gesamtlaufzeit Ihres Algorithmus als Funktion von $n = |L| + |R|$ und $m = |E|$ an. *Provide the overall running time of your algorithm as a function of $n = |L| + |R|$ and $m = |E|$.*

Antwort: $O\left(\underline{(m + n) \cdot \min(n, m)} \right)$.

(/ 10 Punkte)

a) Betrachten Sie die folgende Funktion `f(T)`, die T Einfüge-Operationen `insert`(x) auf einer anfangs leeren Datenstruktur durchführt: *Consider the following function* `f(T)` *that performs T insert operations* `insert`(x) *on an initially empty data structure:*

```
function f(T) {
  for each x in [1, 2, ..., T]:
    insert(x)
}
```

Wir treffen folgende Annahme über `insert`: *We make the following assumption on* `insert`*:*

(A_1) `insert` hat amortisierte Laufzeit $\Theta(a)$ und worst-case Laufzeit $\Theta(a^3)$, wobei a die Anzahl der bisher eingefügten Elemente ist.

(A_1) `insert` *has amortized runtime $\Theta(a)$ and worst-case runtime $\Theta(a^3)$, where a is the number of elements inserted so far.*

Geben Sie unter der Annahme (A_1) die **worst-case** Laufzeit der Funktion `f(T)` in Θ-Notation als Funktion von T an. *Under assumption (A_1), provide the **worst-case** runtime of the function* `f(T)` *in Θ-notation as a function of T.*

Antwort: $\Theta\Big(\underline{\quad T^2 \quad}\Big)$.

(/ 2 Punkte)

b) Betrachten Sie wieder `f(T)` von oben. Diesmal treffen wir folgende Annahme: *Consider* `f(T)` *from above again. This time we make the following assumption:*

(A_2) Die Operation `insert` ist durch einen randomisierten Algorithmus implementiert, der erwartete Laufzeit $\Theta(a)$ und worst-case Laufzeit $\Theta(a^3)$ hat, wobei a die Anzahl der bereits eingefügten Elemente ist.

(A_2) The operation `insert` *is implemented by a randomized algorithm that has expected runtime $\Theta(a)$ and worst-case runtime $\Theta(a^3)$, where a is the number of elements inserted so far.*

Geben Sie unter der Annahme (A_2) die **erwartete** Laufzeit der Funktion `f(T)` in Θ-Notation als Funktion von T an. *Under assumption (A_2), provide the **expected** runtime of the function* `f(T)` *in Θ-notation as a function of T.*

Antwort: $\Theta\Big(\underline{\quad T^2 \quad}\Big)$.

(/ 2 Punkte)

c) Betrachten Sie wieder `f(T)` von oben. Diesmal treffen wir folgende Annahme: *Consider* `f(T)` *from above again. This time we make the following assumption:*

(A_3) Die Operation `insert` ist durch einen randomisierten Algorithmus implementiert. Wenn `insert` aufgerufen wird, nachdem bereits $a \geq 0$ Elemente eingefügt wurden, so ist die Laufzeit des Aufrufs konstant mit Wahrscheinlichkeit $1 - \frac{1}{a+1}$ und sie ist $\Theta(a)$ mit Wahrscheinlichkeit $\frac{1}{a+1}$.

(A_3) The operation `insert` *is implemented by a randomized algorithm. When* `insert` *is called after $a \geq 0$ elements have already been inserted, the runtime of the call is constant with probability $1 - \frac{1}{a+1}$ and it is $\Theta(a)$ with probability $\frac{1}{a+1}$.*

Geben Sie unter der Annahme (A_3) die **erwartete** Laufzeit der Funktion `f(T)` in Θ-Notation als Funktion von T an. *Under assumption (A_3), provide the **expected** runtime of the function* `f(T)` *in Θ-notation as a function of T.*

Antwort: $\Theta\Big(\underline{\quad T \quad}\Big)$.

(/ 2 Punkte)

d) Seien S und R positive natürliche Zahlen und $p \in (0,1)$ gegeben. Wir betrachten folgendes Spiel: S Spieler werfen jeweils eine Münze. Die Münzen sind unabhängig voneinander und zeigen Kopf mit Wahrscheinlichkeit p und Zahl mit Wahrscheinlichkeit $1-p$, wobei p eine Konstante ist. Falls genau ein Spieler Kopf wirft, erhält dieser einen Punkt, ansonsten erhält niemand einen Punkt.

Geben Sie eine Formel für die Wahrscheinlichkeit an, dass Spieler 1 einen Punkt erhält.

Let S and R be positive integers and let $p \in (0,1)$ be given. We consider the following game: S players each flip a coin. The coins are independent of each other and show heads with probability p and tails with probability $1-p$, where p is a constant. If exactly one player flips heads, that player receives a point, otherwise no one receives a point.

Provide a formula for the probability that player 1 receives a point.

Antwort: $p \cdot (1-p)^{S-1}$

Wir nehmen nun an, dass das Spiel insgesamt für R Runden unabhängig wiederholt wird. Geben Sie eine Formel für die Wahrscheinlichkeit an, dass Spieler 1 mindestens einen Punkt erhält.

We now assume that the game is repeated independently for a total of R rounds. Provide a formula for the probability that player 1 receives at least one point.

Antwort: $1 - \left(1 - p \cdot (1-p)^{S-1}\right)^R$

Lösungsskizze: (Es handelt sich um die Zufallsvariablen $S_{1,1}$ und $F_{1,R}$, die in der Vorlesung zu „contention resolution“ analysiert wurden, siehe „Randomisierte Algorithmen I“, Folie 12f.)

(/ 4 Punkte)

e) Angenommen, wir führen eine Folge von n Operationen auf einer Datenstruktur durch, wobei die i-te Operation Kosten von i^2 verursacht, wenn i eine Zweierpotenz ist, und sonst Kosten von 1. Beweisen Sie mit der Aggregationsmethode (=Summenmethode), dass die amortisierten Kosten pro Operation höchstens $5n$ sind. (Erinnerung: Es gilt $\sum_{j=1}^{\ell} b^j \leq b^{\ell+1}$ für alle $b \geq 2$ und $\ell \in \mathbb{N}$.)

Suppose we perform a sequence of n operations on a data structure in which the ith operation costs i^2 if i is an exact power of 2, and 1 otherwise. Use aggregate analysis (=summation method) to prove that the amortized cost per operation is at most $5n$. (Reminder: We have $\sum_{j=1}^{\ell} b^j \leq b^{\ell+1}$ for all $b \geq 2$ and $\ell \in \mathbb{N}$.)

Lösungsskizze: (Es handelt sich um eine Routine-Anwendung der Aggregationsmethode, siehe zum Beispiel Übungsaufgabe 4.1a.)
Aggregationsmethode Sei c_i die Kosten der i-ten Operation.

$$c_i = \begin{cases} i^2\,, & \text{wenn } i \text{ eine Zweierpotenz ist}\,, \\ 1\,, & \text{sonst}\,. \end{cases}$$

Die Kosten von n Operationen betragen:

$$\sum_{i=1}^{n} c_i \leq n + \sum_{j=0}^{\lfloor \log_2 n \rfloor} (2^j)^2 = n + \sum_{j=0}^{\lfloor \log_2 n \rfloor} 4^j \leq n + 4^{\lfloor \log_2 n \rfloor + 1} \leq n + 4^{\log_2 n + 1} = n + 4n^2 \leq 5n^2\,.$$

Daher sind die Durchschnittskosten pro Operation $\leq 5n^2/n = 5n$. Nach der Aggregationsmethode sind die amortisierten Kosten der Operation also $\leq 5n$.

(/ 5 Punkte)

f) Sei G ein ungerichteter Graph mit n Knoten und m Kanten. Sei (A, B) ein minimaler Schnitt von G, also eine Partition in zwei nichtleere Knotenmengen, die möglichst wenige kreuzende Kanten haben. Sei F die Menge der Kanten, die diesen Schnitt kreuzen.

Beweisen Sie zunächst, dass jeder Knoten $v \in V(G)$ mindestens den Grad $|F|$ hat.

Let G be an undirected graph with n nodes and m edges. Let (A, B) be a minimum cut of G, that is, a partition into two non-empty sets of nodes that have as few crossing edges as possible. Let F be the set of edges that cross this cut.

First, prove that every node $v \in V(G)$ has degree at least $|F|$.

Lösungsskizze: (Diese Aufgabe wurde im Vorlesungsvideo behandelt, siehe „Randomisierte Algorithmen I“, Folie 23.)
Sei v ein Knoten von G. Falls $\deg(v) < |F|$, dann wäre $(\{v\}, V(G) - \{v\})$ ein Schnitt mit weniger als $|F|$ kreuzenden Kanten, was ein Widerspruch zur Minimalität von F ist.

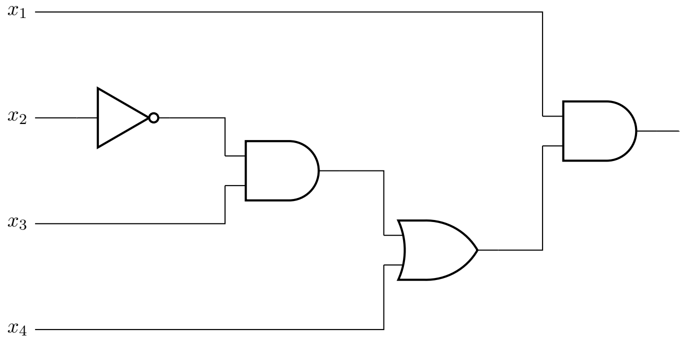
Wir wählen nun eine Kante $e \in E(G)$ zufällig gleichverteilt aus. Zeigen Sie, dass die Wahrscheinlichkeit, dass e den Schnitt (A, B) kreuzt, höchstens $\frac{2}{n}$ ist.

We now sample an edge $e \in E(G)$ uniformly at random. Show that the probability that e crosses the cut (A, B) is at most $\frac{2}{n}$.

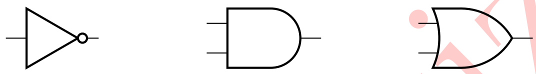
Lösungsskizze: (Diese Aufgabe wurde im Vorlesungsvideo behandelt, siehe „Randomisierte Algorithmen I“, Folie 23.)
Die Kante e kreuzt (A, B) genau dann, wenn $e \in F$ gilt. Wir haben also $\Pr(e \in F) = \frac{|F|}{m}$. Da jeder Knoten von G mindestens den Grad $|F|$ hat, gilt mit dem Handschlaglemma $2m = \sum_{v \in V(G)} \deg(v) \geq n \cdot |F|$. Also ist $\Pr(e \in F) = \frac{|F|}{m} \leq \frac{|F|}{|F|n/2} = \frac{2}{n}$.

(/ 5 Punkte)

a) In der Vorlesung haben wir eine Reduktion von CircuitSat auf 3Sat gesehen. Welche 3Sat-Formel gibt diese Reduktion aus, wenn die Eingabe aus folgendem Schaltkreis besteht? *In the lecture, we have seen a reduction from CircuitSat to 3Sat. Which 3Sat-formula does this reduction output when given as input the following circuit?*



Zur Erinnerung hier die Bedeutung der einzelnen Gatter: *Recall that the meaning of the gates in the above circuit is as follows:*



NICHT-Gatter *NOT gate* | UND-Gatter *AND gate* | ODER-Gatter *OR gate*

Lösungsskizze: (Entspricht Übungsaufgabe 7.1 aus der Vorlesung.)
Wir verwenden weiterhin die Variablen x_i für $1 \leq i \leq 4$ und außerdem eine Variable für den Ausgang jedes Gatters. Diese Ausgänge benennen wir von links nach rechts mit g_1, g_2, g_3, g_4.
Es ergibt sich zunächst die folgende Formel:

$$(g_1 \leftrightarrow \neg x_2) \wedge (g_2 \leftrightarrow (g_1 \wedge x_3)) \wedge (g_3 \leftrightarrow (g_2 \vee x_4)) \wedge (g_4 \leftrightarrow (x_1 \wedge g_3)) \wedge g_4.$$

Daraus ergibt sich durch passende Umformung die folgende 3CNF-Formel:

$$\begin{aligned}
&(\neg g_1 \vee \neg x_2) \wedge (x_2 \vee g_1) \wedge (\neg g_2 \vee g_1) \wedge (\neg g_2 \vee x_3) \wedge (\neg g_1 \vee \neg x_3 \vee g_2) \wedge \\
&(\neg g_3 \vee g_2 \vee x_4) \wedge (\neg g_2 \vee g_3) \wedge (\neg x_4 \vee g_3) \wedge \\
&(\neg g_4 \vee x_1) \wedge (\neg g_4 \vee g_3) \wedge (\neg x_1 \vee \neg g_3 \vee g_4) \wedge g_4
\end{aligned}$$

Diese Formel kann mit Hilfe von zwei zusätzlichen Hilfsvariablen auch leicht in eine Formel mit *genau* drei Literalen pro Klausel umgeformt werden.

(/ 3 Punkte)

b) In der Vorlesung haben wir eine Reduktion von 3Sat auf 3Color gesehen. Welche 3Color-Instanz gibt diese Reduktion aus, wenn die Eingabe die folgende 3CNF-Formel φ ist? Welche Teile des entstehenden Graphen entsprechen den einzelnen Variablen und Klauseln von φ? *In the lecture we have seen a reduction from 3Sat to 3Color. Which instance of 3Color does this reduction output when given as input the following 3CNF-formula φ? What parts of the resulting graph correspond to the individual variables and clauses of φ?*

$$\varphi := (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

Lösungsskizze: (Es handelt sich um eine Anwendung der Reduktion aus der Vorlesung, analog zu Übungsaufgabe 7.1.)
Es ergibt sich der Graph $G = (V, E)$ mit

$$V = \{t, f, v, x_1, \neg x_1, x_2, \neg x_2, x_3, \neg x_3\} \cup \{C_{i,j} \mid 1 \leq i \leq 2, 1 \leq j \leq 5\}$$

und $E = E_1 \cup E_2 \cup E_3$, wobei

$$\begin{aligned}
E_1 &= \{\{t, f\}, \{f, v\}, \{v, t\}\} \cup \{\{v, x_i\}, \{v, \neg x_i\}, \{x_i, \neg x_i\} \mid 1 \leq i \leq 3\},\\
E_2 &= \{\{x_1, C_{1,1}\}, \{\neg x_2, C_{1,2}\}, \{x_3, C_{1,5}\}, \{\neg x_1, C_{2,1}\}, \{x_2, C_{2,2}\}, \{x_3, C_{2,5}\} \text{ und}\\
E_3 &= \{\{C_{i,1}, C_{i,2}\}, \{C_{i,1}, C_{i,3}\}, \{C_{i,2}, C_{i,3}\} \mid 1 \leq i \leq 2\} \cup\\
&\quad \{\{C_{i,3}, C_{i,4}\}, \{C_{i,4}, C_{i,5}\}, \{C_{i,4}, t\}, \{C_{i,5}, t\} \mid 1 \leq i \leq 2\}.
\end{aligned}$$

Dabei korrespondieren die Knoten, deren Namen Literale sind, zu den jeweiligen Variablen und die Knoten $C_{i,1}$ bis $C_{i,5}$ jeweils zur i-ten Klausel.

(/ 3 Punkte)

c) Sei Forest das Entscheidungsproblem, für einen gegebenen ungerichteten Graphen festzustellen, ob er kreisfrei ist. Für welche zwei der folgenden fünf Aussagen ist bereits bekannt, ob sie wahr sind? *Let Forest be the decision problem asking whether a given undirected graph is acyclic. For which two of the following five statements is it already known whether they are true?*

☐ Wenn es eine Polynomialzeitreduktion von 3Sat auf Clique gibt, dann folgt unmittelbar P = NP. *If there is a polynomial time reduction from 3Sat to Clique, then it immediately follows that* P = NP.

☑ Wenn es eine Polynomialzeitreduktion von Clique auf Forest gibt, dann folgt unmittelbar P = NP. *If there is a polynomial time reduction from Clique to Forest, then it immediately follows that* P = NP.

☐ Wenn es eine Polynomialzeitreduktion von Forest auf 3Sat gibt, dann folgt unmittelbar P = NP. *If there is a polynomial time reduction from Forest to 3Sat, then it immediately follows that* P = NP.

☐ Wenn P = NP gilt, dann gibt es keine Polynomialzeitreduktion von 3Sat auf Forest. *If* P = NP *holds, then there is no polynomial time reduction from 3Sat to Forest.*

☑ Wenn P $\neq$ NP gilt, dann gibt es keine Polynomialzeitreduktion von 3Sat auf Forest. *If* P $\neq$ NP *holds, then there is no polynomial time reduction from 3Sat to Forest.*

(/ 2 Punkte)

d) Seien Π und Π' zwei Entscheidungsprobleme. Welche zwei der folgenden fünf Aussagen sind wahr? *Let Π and Π' be two decision problems. Which two of the following five statements are true?*

☐ Um zu beweisen, dass das Problem Π' NP-schwer ist, genügt es, zu beweisen, dass Π in NP ist und es eine Polynomialzeitreduktion von Π nach Π' gibt. *In order to prove that Π' is NP-hard, it suffices to show that Π is in NP and there is a polynomial time reduction from Π to Π'.*

☑ Um zu beweisen, dass Π' in NP ist, genügt es, zu beweisen, dass Π in NP ist und es eine Polynomialzeitreduktion von Π' nach Π gibt. *In order to prove that Π' is in NP, it suffices to show that Π is in NP and there is a polynomial time reduction from Π' to Π.*

☐ Um zu beweisen, dass Π' NP-vollständig ist, genügt es, zu beweisen, dass Π NP-vollständig ist und es eine Polynomialzeitreduktion von Π nach Π' gibt. *In order to prove that Π' is NP-complete, it suffices to show that Π is NP-complete and there is a polynomial time reduction from Π to Π'.*

☐ Wenn Π NP-schwer ist und es eine Polynomialzeitreduktion von Π' nach Π gibt, dann folgt direkt, dass Π' NP-schwer ist. *If Π is NP-hard and there is a polynomial time reduction from Π' to Π, then it immediately follows that Π' is NP-hard.*

☑ Wenn Π in NP ist und es keine Polynomialzeitreduktion von Π auf das Problem $\{ G \mid G$ ist ungerichteter Graph ohne Kanten $\}$ gibt, dann folgt direkt P $\neq$ NP. *If Π is in NP and there is no polynomial time reduction from Π to the problem $\{ G \mid G$ is an undirected graph without edges $\}$ then it immediately follows that* P $\neq$ NP.

(/ 2 Punkte)

e) Wir betrachten das folgende Entscheidungsproblem DirectedHalfiltonianCycle:

Eingabe:	Gerichteter Graph $G = (V, E)$.
Frage:	Gibt es in G einen einfachen Kreis mit mindestens $|V|/2$ Knoten?

Beweisen Sie, dass das Problem DirectedHalfiltonianCycle NP-schwer ist.

Consider the following decision problem DirectedHalfiltonianCycle:

Input:	*Directed graph $G = (V, E)$.*
Question:	*Is there a simple cycle in G that contains at least $|V|/2$ nodes?*

Prove that the problem DirectedHalfiltonianCycle is NP-hard.

Lösungsskizze: (Es handelt sich um eine Variante von Übungsaufgabe 8.2a.) Reduktion von DirectedHamiltonianCycle. Sei $G = (V, E)$ die Eingabe für die Reduktion. Sei $G' = (V', E)$ der Graph G mit $|V|$ zusätzlichen isolierten Knoten. Offenbar lässt sich G' in polynomieller Zeit berechnen. Nun gilt:

$$\begin{aligned}
G \in \textsc{DirectedHamiltonianCycle} &\iff G \text{ enthält einen einfachen Kreis mit } |V| \text{ Knoten} \\
&\iff G' \text{ enthält einen einfachen Kreis mit } |V| \text{ Knoten} \\
&\iff G' \text{ enthält einen einfachen Kreis mit } \geq |V'|/2 \text{ Knoten} \\
&\iff G' \in \textsc{DirectedHalfiltonianCycle}
\end{aligned}$$

(/ 10 Punkte)

Sei $M_{\text{agic}} := (Q, \Sigma, \Gamma, \delta, q_0, \texttt{acc}, \texttt{rej}, \square)$ eine Turingmachine, wobei $Q := \{q_i \mid 0 \leq i \leq 3\}$, $\Sigma := \{a, b, c\}$, $\Gamma := \Sigma \cup \{\square\}$ und die Überführungsfunktion δ durch die folgenden Übergänge gegeben ist: *Let $M_{agic} := (Q, \Sigma, \Gamma, \delta, q_0, \texttt{acc}, \texttt{rej}, \square)$ be a Turing machine, where $Q := \{q_i \mid 0 \leq i \leq 3\}$, $\Sigma := \{a, b, c\}$, $\Gamma := \Sigma \cup \{\square\}$ and the transition function δ is given by the following transitions:*

$(q_0, c) \to (q_1, c, +1)$ $\quad$ $(q_1, \square) \to (\texttt{acc}, \square, -1)$
$(q_1, a) \to (q_2, a, -1)$ $\quad$ $(q_2, b) \to (q_3, b, +1)$
$(q_1, b) \to (q_1, b, +1)$ $\quad$ $(q_3, a) \to (q_3, a, +1)$
$(q_1, c) \to (q_1, c, +1)$ $\quad$ $(q_3, b) \to (q_1, b, +1)$

a) Führen Sie die Maschine M_{agic} auf Eingabe *cbaaabcbaa* aus, bis die Maschine akzeptiert oder abstürzt. Geben Sie in geeigneter Weise die Startkonfiguration und die Konfigurationen nach jedem Berechnungsschritt an, also den Bandinhalt, die Kopfposition und den aktuellen Zustand. *Run the machine M_{agic} on input cbaaabcbaa until the machine either accepts or crashes. State the starting configuration and the configurations after each step of the computation in a suitable manner, that is, state the content of the tape, the position of the head, and the current state in each of the steps.*

Lösungsskizze: (Es handelt sich um eine Variante von Übungsaufgabe 9.1a.)

Zustand	Bandinhalt (Kopfposition unterstrichen)
q_0	$\underline{c}baaabcbaa$
q_1	$c\underline{b}aaabcbaa$
q_1	$cb\underline{a}aabcbaa$
q_2	$c\underline{b}aaabcbaa$
q_3	$cb\underline{a}aabcbaa$
q_3	$cba\underline{a}abcbaa$
q_3	$cbaa\underline{a}bcbaa$
q_3	$cbaaa\underline{b}cbaa$
q_1	$cbaaab\underline{c}baa$
q_1	$cbaaabc\underline{b}aa$
q_1	$cbaaabcb\underline{a}a$
q_2	$cbaaabc\underline{b}aa$
q_3	$cbaaabcb\underline{a}a$
q_3	$cbaaabcba\underline{a}$
q_3	$cbaaabcbaa\underline{\square}$

(/ 4 Punkte)

b) Geben Sie die Sprache an, die von M_{agic} entschiedene Sprache an. *What language does the machine M_{agic} decide?*

Lösungsskizze:

$$\left\{ w \in \{a, b, c\}^+ \;\middle|\; \begin{array}{c} w \text{ beginnt mit } c \text{ und jeder Block von } a\text{'s in } w \\ \text{ist auf beiden Seiten von mindestens} \\ \text{je einem } b \text{ umgeben} \end{array} \right\}$$

(/ 2 Punkte)

c) Sei $\Sigma := \{0,1\}$. Welche zwei der folgenden fünf Sprachen sind entscheidbar?
Let $\Sigma := \{0,1\}$. Which two of the following five languages are decidable?

- ☐ $\{\langle M, x, y\rangle \mid M$ hält auf Eingabe x und hält nicht auf Eingabe $y\}$
  $\{\langle M, x, y\rangle \mid M$ halts on input x and does not halt on input $y\}$
- ☑ $\{\langle M\rangle \mid M$ hält auf allen Eingaben in höchstens 10 Schritten$\}$
  $\{\langle M\rangle \mid M$ halts on all inputs in at most 10 steps$\}$
- ☑ $\{\langle M\rangle \mid M$ überschreibt auf mindestens einer Eingabe ein Symbol mit einem anderen Symbol $\}$
  $\{\langle M\rangle \mid M$ replaces some symbol on the tape by a different symbol on at least one input$\}$
- ☐ $\{\langle M\rangle \mid M$ hält auf allen Eingaben gerader Länge$\}$
  $\{\langle M\rangle \mid M$ halts on all inputs of even length$\}$
- ☐ $\{\langle M, x\rangle \mid M$ besucht auf Eingabe x die erste Bandzelle mindestens zwei mal$\}$
  $\{\langle M, x\rangle \mid M$ on input x visits the first cell of the tape at least twice$\}$

(/ 2 Punkte)

d) Welche zwei der folgenden fünf Aussagen sind wahr? *Which two of the following five statements are true?*

- ☑ Für alle Sprachen L gilt: Sind L und das Komplement $\overline{L}$ semi-entscheidbar, so ist $\overline{L}$ entscheidbar. *For all languages L the following holds: If L and its complement $\overline{L}$ are semidecidable, then $\overline{L}$ is decidable.*
- ☐ Für alle Sprachen L und L' gilt: Wenn $L \subseteq L'$, so gilt auch $L \leq L'$. *For all languages L and L' the following holds: If $L \subseteq L'$, then also $L \leq L'$.*
- ☐ Für alle Sprachen L und L' gilt: Wenn $L \leq L'$, so gilt auch $L \subseteq L'$. *For all languages L and L' the following holds: If $L \leq L'$ then also $L \subseteq L'$.*
- ☐ Für jede Sprache L gilt: Entweder ist L semi-entscheidbar oder das Komplement $\overline{L}$ ist semi-entscheidbar. *For all languages L the following holds: Either L is semidecidable or its complement $\overline{L}$ is semidecidable.*
- ☑ Für alle Sprachen L und L' gilt: Wenn es eine Reduktion von L nach L' gibt und L nicht semi-entscheidbar ist, dann ist auch L' nicht semi-entscheidbar. *For all languages L and L' the following holds: If there is a reduction from L to L' and L is semidecidable, then L' is semidecidable as well.*

(/ 2 Punkte)

e) Zeigen Sie mit dem Satz von Rice, dass die folgende Sprache nicht entscheidbar ist:

$$L \coloneqq \{\langle M\rangle \mid M \text{ hält auf mindestens einer Eingabe jeder Eingabelänge}\}.$$

Use Rice's theorem to show that the following language is not decidable:

$$L \coloneqq \{\langle M\rangle \mid M \textit{ halts on at least one input of each input length}\}.$$

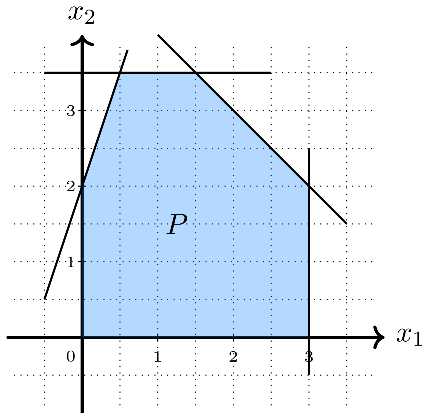
Lösungsskizze: (Es handelt sich um eine Variante von Übungsaufgabe 10.3b.)
Satz von Rice mit

$$\mathcal{S} \coloneqq \left\{ f \;\middle|\; \begin{array}{c} f \text{ ist berechenbare Funktion, die auf mindestens} \\ \text{einer Eingabe jeder Eingabelänge definiert ist} \end{array} \right\}.$$

Die überall undefinierte Funktion Ω ist nicht in $\mathcal{S}$, während eine überall definierte Funktion in $\mathcal{S}$ ist. Somit ist $\mathcal{S}$ nicht trivial. Weiterhin ist $C(\mathcal{S}) = L$ und somit liefert der Satz von Rice, dass L nicht entscheidbar ist.

(/ 10 Punkte)

a) Gegeben sei folgendes Polytop P in $\mathbb{R}^2$: *Consider the following polytope P in $\mathbb{R}^2$:*



Sei das lineare Programm (LP_1) gegeben durch $\max x_1 + x_2$ unter der Nebenbedingung $(x_1, x_2) \in P$. Geben Sie eine optimale zulässige Lösung x^* für (LP_1) an.
Let the linear program (LP_1) be given by $\max x_1 + x_2$ subject to $(x_1, x_2) \in P$. State an optimal feasible solution x^* for (LP_1):

Antwort: $x_1^* =$ 3, $x_2^* =$ 2.

(/ 2 Punkte)

b) Welche zwei der folgenden fünf Aussagen über (LP_1) aus Aufgabenteil a) sind wahr? *Which two of the following five statements about (LP_1) from subtask a) are true?*

☑ Jede optimale Lösung des dualen Programms von (LP_1) hat einen Zielfunktionswert von 5. *Every optimal solution of the dual program of (LP_1) has an objective function value of 5.*

☑ Das duale Programm von (LP_1) hat 4 Variablen und 2 Constraints. *The dual program of (LP_1) has 4 variables and 2 constraints.*

☐ Der Vektor $(3, 0)$ ist eine optimale zulässige Lösung für (LP_1). *The vector $(3, 0)$ is an optimal feasible solution for (LP_1).*

☐ Eine mögliche Reihenfolge von Knoten, die der Simplex-Algorithmus auf der Suche nach einer optimalen Lösung für (LP_1) betrachten kann, ist durch $(0, 0), (0, 2), (0, 0), (3, 0)$ gegeben. *A possible order of nodes that the simplex algorithm can consider in search of an optimal solution for (LP_1) is given by $(0, 0), (0, 2), (0, 0), (3, 0)$.*

☐ Das duale Programm von (LP_1) hat 2 Variablen und 6 Constraints. *The dual program of (LP_1) has 2 variables and 6 constraints.*

(/ 2 Punkte)

c) Betrachten Sie folgendes, in kanonischer Form gegebene lineare Programm (LP_2):

$$\begin{array}{lrl} \text{maximize} & 7x + 6y - z & \\ \text{subject to} & x + 2y & \leq 80 \\ & y + 9z & \leq 3 \\ & x, y, z & \geq 0 \end{array}$$

Überführen Sie (LP_2) in ein äquivalentes lineares Programm in Standardform.
Consider the linear program (LP_2) above, given in canonical form. Convert it to an equivalent linear program in standard form.

Lösungsskizze: (Es handelt sich um eine Variante von Übungsaufgabe 12.2a.)

$$\begin{array}{lrl} \text{maximize} & 7x + 6y - z & \\ \text{subject to} & x + 2y + s_1 & = 80 \\ & y + 9z + s_2 & = 3 \\ & x, y, z, s_1, s_2 & \geq 0 \end{array}$$

(/ 3 Punkte)

d) Betrachten Sie erneut das LP_2 aus c) in kanonischer Form. Geben Sie das duale Programm von LP_2 an. *Consider LP_2 from c) again. State the dual program of LP_2.*

Lösungsskizze: (Es handelt sich um eine Variante von Übungsaufgabe 13.1.)

$$\begin{array}{lrl} \text{minimize} & 80u + 3v & \\ \text{subject to} & u & \geq 7 \\ & 2u + v & \geq 6 \\ & 9v & \geq -1 \\ & u, v & \geq 0 \end{array}$$

(/ 3 Punkte)

e) Sei $C \in \mathbb{N}$ eine positive Konstante. Wir definieren das folgende Optimierungsproblem.

Maximale Teilsumme

Eingabe: Ein Array $A[1,\dots,n]$ von ganzen Zahlen aus der Menge $\{1,\dots,C\}$.
Ausgabe: Der Wert der maximalen Teilsumme,
also die folgende Zahl: $\mathrm{opt}(A) := \max_{S\subseteq\{1,\dots,n\}} \sum_{i\in S} A[i]$.

Betrachten Sie den folgenden Algorithmus `grob(A)`.

```
function grob(A[1,...,n]) {
    i = 1
    s = 0
    while (i <= n) {
        s = s + A[i]
        i = i + 2
    }
    return s
}
```

Zeigen Sie, dass `grob` ein $7C$-Approximationsalgorithmus für Maximale Teilsumme ist. (Hinweis: Diese Aufgabe ist deutlich einfacher als sie zunächst aussieht.)

Let $C \in \mathbb{N}$ be a positive constant. We define the following optimization problem.

Maximum Subsum

Input: *An array $A[1,\dots,n]$ of integers from the set $\{1,\dots,C\}$.*
Output: *The value of the maximum subsum,*
that is, the following number: $\mathrm{opt}(A) := \max_{S\subseteq\{1,\dots,n\}} \sum_{i\in S} A[i]$.

Show that `grob` stated above is a $7C$-approximation algorithm for Maximum Subsum. (Hint: This task is much easier than it first appears.)

Lösungsskizze: (Hier muss man lediglich die Definition eines Approximationsalgorithmus für ein Maximierungsproblem korrekt anwenden.)
Da alle Zahlen positiv sind, ist die Lösung des Optimierungsproblems immer durch $S = \{1,\dots,n\}$ gegeben, und die maximale Teilsumme ist $A[1] + \cdots + A[n]$. Um zu zeigen, dass `grob` ein $7C$-Approximationsalgorithmus für Maximale Teilsumme ist, genügt es also folgendes zu zeigen:

$$\texttt{grob}(A) \geq \frac{1}{7C} \cdot (A[1] + \cdots + A[n])\,.$$

Zunächst beobachten wir, dass $A[1] + \cdots + A[n] \leq Cn$ gilt, da alle Einträge von A in $\{1,\dots,C\}$ liegen. Der Algorithmus `grob` berechnet die Summe $\texttt{grob}(A) = A[1] + A[3] + A[5] + \dots$, nimmt also nur jeden zweiten Summanden. Diese Summe hat genau $\lfloor (n+1)/2 \rfloor \geq \frac{1}{2}n$ Summanden und jeder Summand ist mindestens 1. Somit gilt $\texttt{grob}(A) \geq \frac{1}{2}n \cdot 1 = \frac{1}{2}n$.
Zusammengenommen erhalten wir

$$\texttt{grob}(A) \geq \tfrac{1}{2}n \geq \tfrac{1}{7}n \geq \tfrac{1}{7C}(A[1] + \cdots + A[n])\,.$$

Das war zu zeigen.

(/ 10 Punkte)

Wichtig: Lösungen auf dieser Seite werden nur dann berücksichtigt, wenn bei der entsprechenden Aufgabe ein Verweis zu Seite 20 platziert wurde.

Important: *Solutions on this page will only be considered if a reference to page 20 was placed at the corresponding task.*

Wichtig: Lösungen auf dieser Seite werden nur dann berücksichtigt, wenn bei der entsprechenden Aufgabe ein Verweis zu Seite 21 platziert wurde.

Important: *Solutions on this page will only be considered if a reference to page 21 was placed at the corresponding task.*

Wichtig: Lösungen auf dieser Seite werden nur dann berücksichtigt, wenn bei der entsprechenden Aufgabe ein Verweis zu Seite 22 platziert wurde.

Important: *Solutions on this page will only be considered if a reference to page 22 was placed at the corresponding task.*

Wichtig: Lösungen auf dieser Seite werden nur dann berücksichtigt, wenn bei der entsprechenden Aufgabe ein Verweis zu Seite 23 platziert wurde.

Important: *Solutions on this page will only be considered if a reference to page 23 was placed at the corresponding task.*

Wichtig: Lösungen auf dieser Seite werden nur dann berücksichtigt, wenn bei der entsprechenden Aufgabe ein Verweis zu Seite 24 platziert wurde.

Important: *Solutions on this page will only be considered if a reference to page 24 was placed at the corresponding task.*