Technische Universität Berlin **SoSe 2018**
Fakultät II, Institut für Mathematik
Geschäftsstelle Matheon, MA 3–1, Dorothea Kiefer
Dr. Torsten Mütze
Sven Jäger, Dr. Frank Lutz, Manuel Radons

Modulklausur zur Computerorientierten Mathematik I+II

11.10.2018, Beginn: 08:15 Uhr, Bearbeitungszeit: 150 Minuten

Nachname, Vorname: _____

Matrikelnummer: _____

Studiengang: _____

Es sind keine Hilfsmittel erlaubt. Bitte nicht mit Bleistift oder mit roter Farbe schreiben!

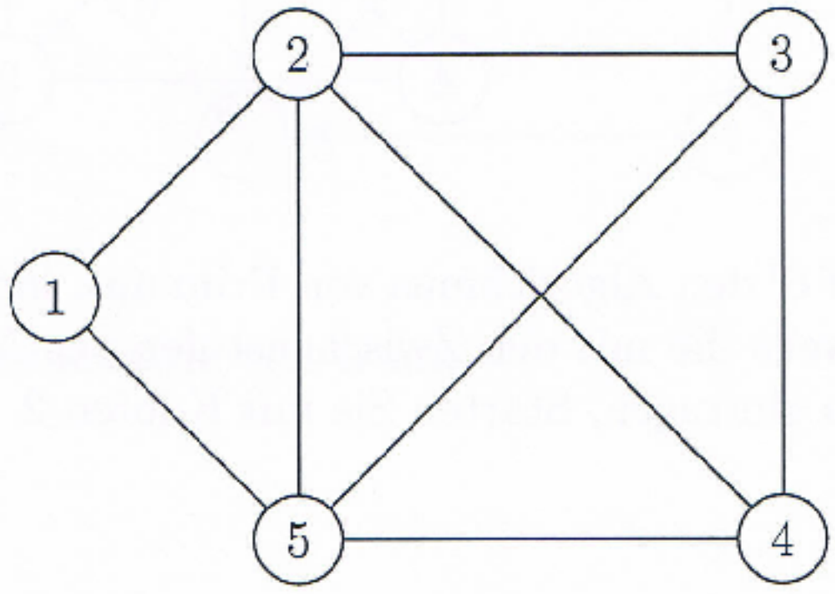
Die Klausur ist mit mindestens ***36*** *(ITM/NidI:* ***30****)* ***Punkten*** *bestanden.*

1	2	3	4	5	6	7	8	Summe
10	14	10	9	13	11	5	8	80

1. Aufgabe (3 + 2 + 5 Punkte)

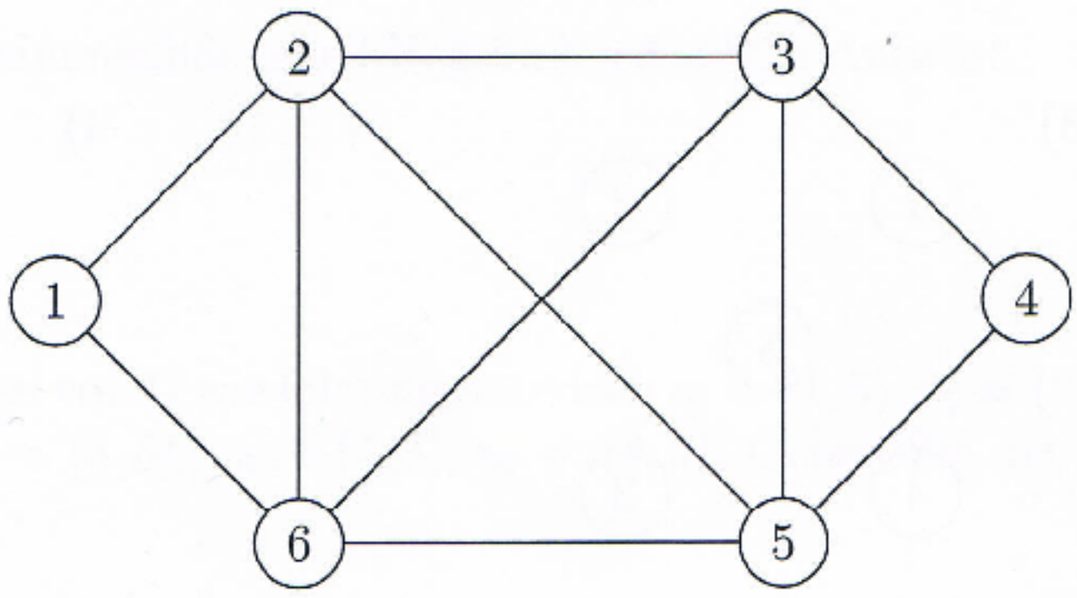
(a) Eine *Eulertour* ist ein geschlossener Kantenzug, der alle Kanten eines Graphen genau einmal enthält. Ein *offener Eulerzug* ist ein nicht notwendig geschlossener Kantenzug, der alle Kanten eines Graphen genau einmal enthält.

(i) Welche möglichen Startknoten für einen offenen Eulerzug hat der folgende ungerichtete Graph?



Antwort: _____

(ii) Begründen Sie, dass der folgende Graph eine Eulertour haben muss.

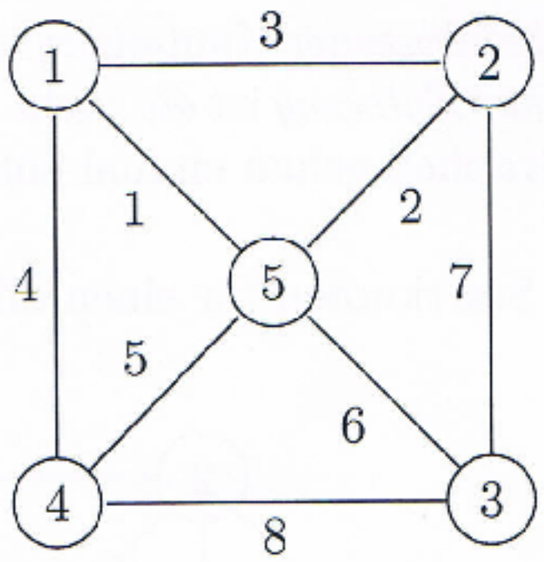


Antwort: _____

Welche Knoten sind mögliche Startknoten?

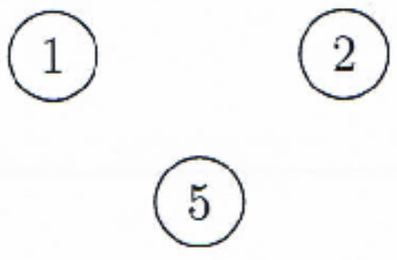
Antwort: _____

(b) Es sei folgender ungerichteter, gewichteter Graph G gegeben.

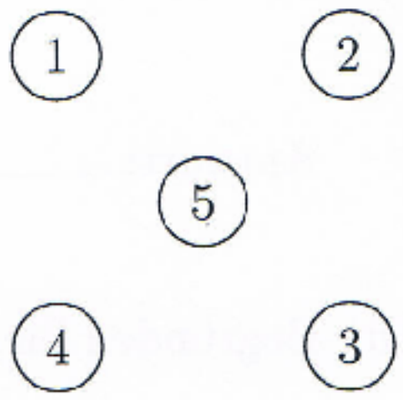


Führen Sie auf G den Algorithmus von Prim aus, indem Sie auf den folgenden Knotenmengen jeweils die mit den Zwischenstufen des Algorithmus korrespondierenden Kantenmengen eintragen. Starten Sie mit Knoten 2.

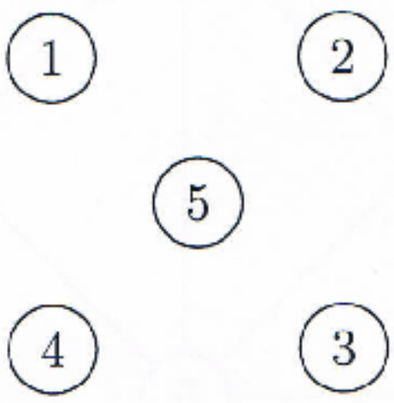
1)



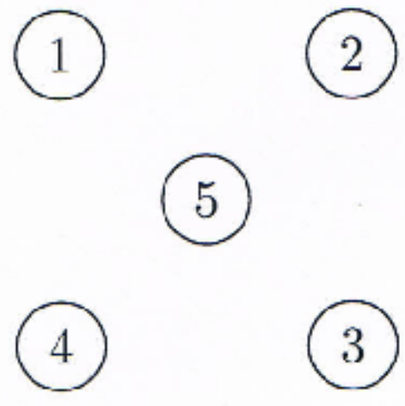
2)



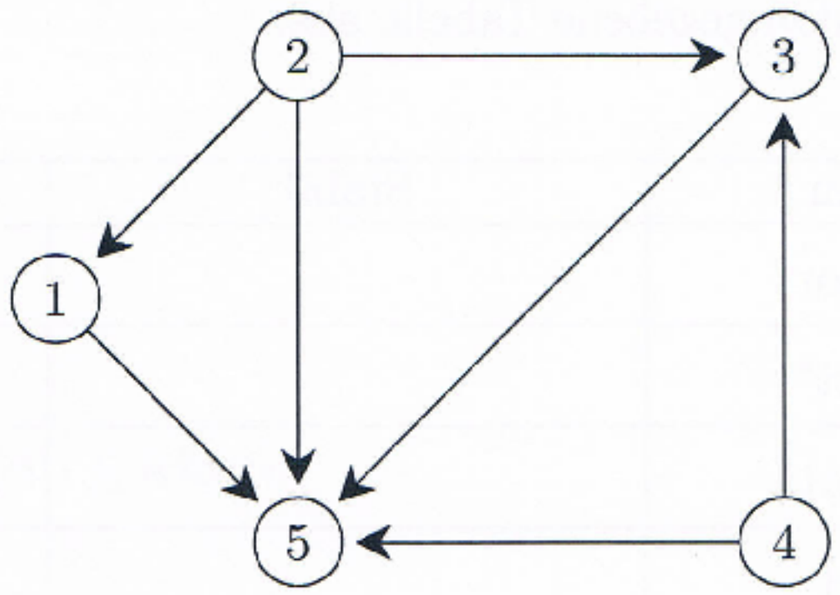
3)



4)



(c) Es sei folgender Digraph G gegeben:



(i) Geben Sie die Adjazenzmatrix von G an:

(ii) Ist G stark zusammenhängend? Begründen Sie Ihre Antwort.

(iii) Seien die Kanten von G wie folgt nummeriert: $e_1 = (1,5)$, $e_2 = (2,1)$, $e_3 = (2,3)$, $e_4 = (2,5)$, $e_5 = (3,5)$, $e_6 = (4,3)$, $e_7 = (4,5)$. Geben Sie die Inzidenzmatrix von G an:

2. Aufgabe

(4 + 4 + 3 + 3 Punkte)

(a) Füllen Sie die unten gegebene Tabelle aus.

Verfahren	Stabil	Inplace
Insertionsort		
Mergesort*		
Bucketsort		
Heapsort		

∗ Implementation aus der Vorlesung

(b) Führen Sie Heapsort aufsteigend mit MaxHeaps für die folgende Liste durch. Machen Sie kenntlich, wo Build-Max-Heap aufhört und wo die eigentliche Sortierung beginnt. Elemente, die aus dem Heap entfernt wurden, müssen nicht erneut aufgeschrieben werden.

$$[8, 7, 12, 4, 2, 6, 9].$$

(c) (i) Formulieren Sie die untere Schranke für vergleichsbasiertes Sortieren als Theorem.

(ii) Beweisen Sie: $\log(n!) \leq n \log(n)$.

(d) Beschreiben Sie, wie man in Linearzeit n vierstellige Dezimalzahlen sortieren kann. Erläutern Sie, wie dies zur unteren Schranke aus Aufgabenteil (c) passt.

3. Aufgabe (2 + 2 + 4 + 2 Punkte)

(a) Sei die Grundmenge $E = \{1, 2\}$ gegeben. Geben Sie alle Unabhängigkeitssysteme auf E an:

(b) Sei $\mathcal{F} \subseteq \mathcal{P}(E)$ ein Unabhängigkeitssystem auf der Grundmenge E. Zeigen oder widerlegen Sie, ob durch die folgende Bedingung an das Tupel $(E, \mathcal{F})$ ein Matroid definiert wird:

(M) Seien B, B' Basen von E, so ist $|B| = |B'|$.

(c) Betrachten Sie das Matroid $(E, \mathcal{I})$, wobei $E = \{a, b, c, d, e\}$ die Menge der Spalten der reellen Matrix

$$A = \begin{array}{c} \begin{matrix} a & b & c & d & e \end{matrix} \\ \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 5 & -2 & 2 & 3 & 3 \\ 0 & 0 & 7 & 0 & 0 \end{pmatrix} \end{array}$$

bezeichne und $\mathcal{I}$ die Menge der linear unabhängigen Teilmengen von E ist. Geben Sie sämtliche Basen und Kreise des Matroids $(E, \mathcal{I})$ an.

(d) Gegeben sei ein Matroid $(E, \mathcal{I})$ mit $E = \{a, b, c, d\}$ und Basismenge

$$B = \{\{a, b, d\}, \{a, c, d\}, \{b, c, d\}\}.$$

Zeichnen Sie alle *kanten-gelabelten* Graphen $G = (V, E)$, so dass $(E, \mathcal{I})$ das graphische Matroid von G ist.

4. Aufgabe (2 + 3 + 4 Punkte)

(a) (i) Beschreiben Sie, was der Begriff primäres Clustering im Kontext des Hashings bedeutet.

(ii) Bei welchem Hashverfahren tritt primäres Clustering insbesondere auf?

(b) Für eine positive ganze Zahl k sei $\mathbf{Z}_k := \{0, 1, \ldots, k-1\}$. Gegeben sind der Datensatz

$$[8, 18, 4, 6, 7, 13, 9]$$

und die Hashfunktion

$$h : \mathbb{Z} \to \mathbf{Z}_7\,, \quad x \mapsto (5x + 2) \mod 7\,.$$

(i) Hashen Sie den Datensatz mit h, wobei Kollisionen mittels Verkettung aufgelöst werden. Geben Sie den Zustand der Hashtabelle **nach** den Einfügungen an:

(ii) Hashen Sie den Datensatz mit h, wobei Kollisionen mittels linearer Sondierung aufgelöst werden. Geben Sie hierbei den Zustand der Hashtabelle **nach jeder Einfügung** an:

(c) Sei $n \in \mathbb{N}_{>0}$ und $R_n := \{1, \ldots, n-1\}$. Seien weiterhin $a, b \in R_n$.

(i) Es sei n eine Primzahl. Zeigen Sie: Es gibt ein $k \in \mathbb{N}_{>0}$, so dass $a^k \bmod n = 1$.

(ii) Es sei $n \in \mathbb{N}_{>0}$ potenzfrei in dem Sinn, dass es keine Zahlen $m, k \in \mathbb{N}_{>1}$ gibt, so dass gilt $n = m^k$. Beantworten Sie folgende Fragen:

- Gibt es notwendigerweise ein $\ell \in \mathbb{N}_{>0}$, so dass gilt $a^\ell \bmod n = 1$?
- Gilt notwendig $a \cdot b \bmod n \in R_n$?
- Muss es ein $c \in R_n$ geben, so dass gilt $a \cdot c \bmod n = 1$?

5. Aufgabe (3 + 3 + 4 + 1 + 2 Punkte)

(a) Führen Sie den Huffman-Algorithmus für das Wort

COOOOMMAAA

durch. Zeichnen Sie die Teilbäume auf den verschiedenen Stufen des Algorithmus und tragen Sie den resultierenden Code in die Tabelle ein.

C	O	M	A

(b) Begründen oder widerlegen Sie, dass die folgenden Codes eindeutig decodierbar sind.

(i)

a	b	c	d	e	f
111	110	010	00	001	10

(ii)

a	b	c	d	e	f
000	01	1000	101	110	111

(iii)

a	b	c	d	e	f
000	001	100	101	110	111

(c) Ein binärer Suchbaum heißt *voll*, falls alle Ebenen bis auf die letzte voll belegt sind.

Sei B ein binärer Suchbaum. Beweisen oder widerlegen Sie die folgenden Aussagen:

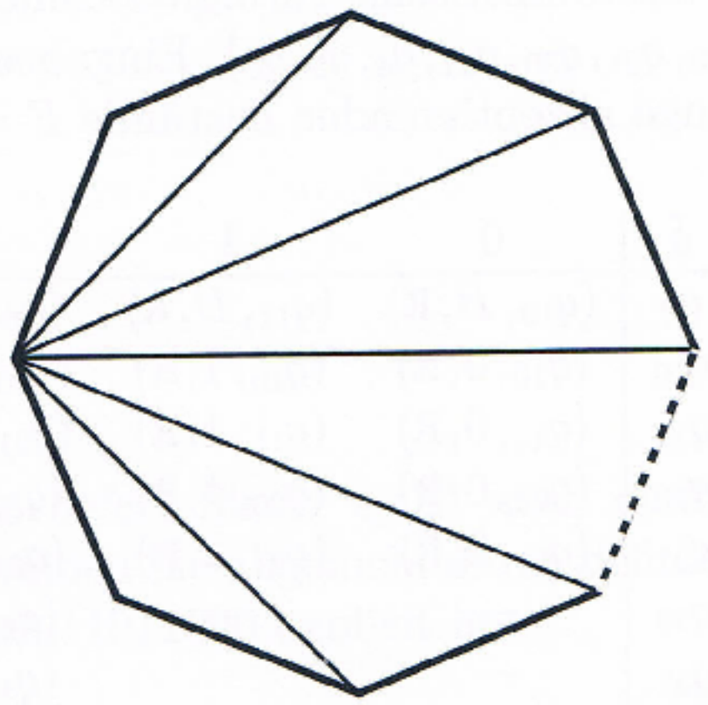
(i) Wenn B voll ist, dann ist B ein AVL-Baum.

(ii) Wenn B ein AVL-Baum ist, dann ist B voll.

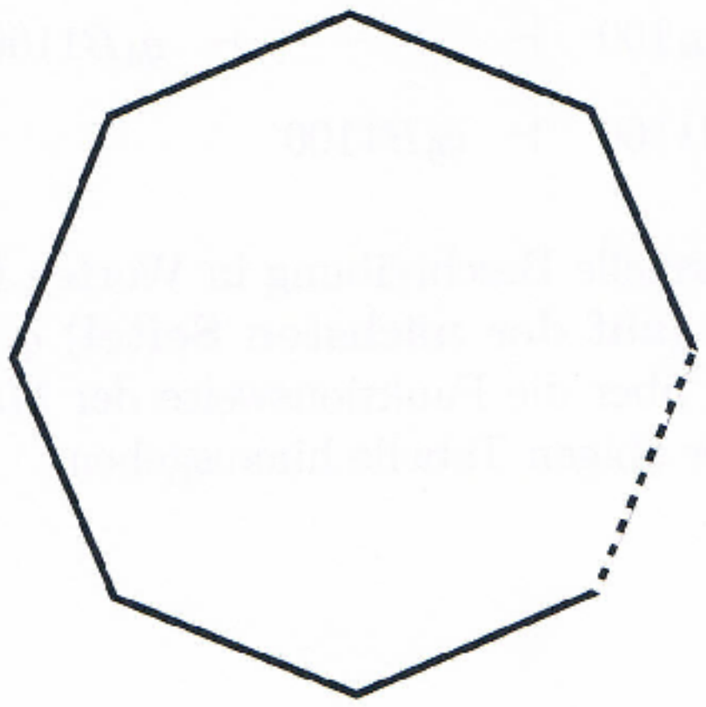
(iii) Wenn je zwei Blätter von B einen Höhenunterschied von höchstens 1 haben, dann ist B ein AVL-Baum.

(iv) Wenn B ein AVL-Baum ist, dann haben je zwei Blätter von B Höhenunterschied höchstens 1.

(d) Zeichnen Sie den dualen Graphen (Binärbaum) der folgenden Triangulierung eines 8-Ecks. Es sei dabei der Wurzelknoten durch die gestrichelte Linie ausgezeichnet.



(e) Triangulieren Sie das folgende 8-Eck so, dass der duale Graph ein voller binärer Baum ist:



6. Aufgabe (4 + 3 + 2 + 2 Punkte)

Gegeben sei die folgende deterministische Turingmaschine mit Startzustand q_0, Zustandsmenge $Q = \{q_0, q_{10}, q_{11}, q_{20}, q_{21}, q_{30}, q_{31}, q_4, q_5, q_6\}$, Eingabealphabet $\Sigma = \{0, 1\}$, Bandalphabet $\Gamma = \Sigma \cup \{B\}$ und Menge akzeptierender Zustände $F = \{q_6\}$:

δ	0	1	B
q_0	$(q_{10}, B, \mathtt{R})$	$(q_{11}, B, \mathtt{R})$	$(q_6, B, \mathtt{N})$
q_{10}	$(q_{10}, 0, \mathtt{R})$	$(q_{10}, 1, \mathtt{R})$	$(q_{20}, B, \mathtt{R})$
q_{11}	$(q_{11}, 0, \mathtt{R})$	$(q_{11}, 1, \mathtt{R})$	$(q_{21}, B, \mathtt{R})$
q_{20}	$(q_{20}, 0, \mathtt{R})$	$(q_{20}, 1, \mathtt{R})$	$(q_{30}, 1, \mathtt{R})$
q_{21}	$(q_{21}, 0, \mathtt{R})$	$(q_{21}, 1, \mathtt{R})$	$(q_{31}, 0, \mathtt{R})$
q_{30}	—	—	$(q_4, 1, \mathtt{L})$
q_{31}	—	—	$(q_4, 0, \mathtt{L})$
q_4	$(q_4, 0, \mathtt{L})$	$(q_4, 1, \mathtt{L})$	$(q_5, B, \mathtt{L})$
q_5	$(q_5, 0, \mathtt{L})$	$(q_5, 1, \mathtt{L})$	$(q_0, B, \mathtt{R})$
q_6	—	—	—

Hier bedeutet ‚—' in Zeile q und Spalte σ den Eintrag $(q, \sigma, \mathtt{N})$.

(a) Vervollständigen Sie die folgende Konfigurationsfolge:

$$
\begin{array}{llllll}
Bq_001 & \vdash & & \vdash & & \vdash\ 1Bq_{20}B \\
1Bq_{30}1B & \vdash & & \vdash & & \vdash\ q_51B11 \\
q_5B1B11 & \vdash & & \vdash & & \vdash\ Bq_{21}11 \\
B1q_{21}1 & \vdash & & \vdash & B110q_{31}B & \vdash\ B11q_400 \\
B1q_4100 & \vdash & & \vdash & q_4B1100 & \vdash\ q_5BB1100 \\
q_0B1100 & \vdash & q_6B1100 & & &
\end{array}
$$

(b) Geben Sie eine informelle Beschreibung in Worten für die Bedeutung der Zustände q_0, q_{10}, q_{20}, q_{30} und (**auf der nächsten Seite!**) q_4 und q_5 an. Diese Beschreibung sollte eine Intuition über die Funktionsweise der Maschine vermitteln und über die Informationen in der obigen Tabelle hinausgehen.

q_0:

q_{10}:

q_{20}:

q_{30}:

q_4:

q_5:

(c) Die obige Turingmaschine liefert für $w \in \Sigma^*$ nach endlich vielen Schritten einen Output $f(w) \in \Sigma^*$. Beschreiben Sie allgemein die so definierte Funktion $f: \Sigma^* \to \Sigma^*$. Geben sie $f(w)$ für $w = 10011011000$ explizit an.

(d) Geben Sie Speicherplatzbedarf und Laufzeit der Maschine asymptotisch in Abhängigkeit von der Eingabelänge n in Θ-Notation an.

- Speicherbedarf:
- Laufzeit:

7. Aufgabe

(1 + 1 + 1 + 1 + 1 Punkte)

Bei den folgenden Teilaufgaben (a)–(e) ist jeweils genau eine der zur Auswahl stehenden Antworten zutreffend. Kennzeichnen Sie die jeweils korrekte Antwort durch das Setzen genau eines Kreuzes pro Teilaufgabe.

(a) Welche der folgenden Aussagen ist **falsch**?

- ☐ Jeder Präfixcode ist eindeutig decodierbar.
- ☐ Jeder Blockcode ist ein Präfixcode.
- ☐ Jeder Präfixcode lässt sich mit einem AVL-Baum identifizieren.
- ☐ Jeder Präfixcode lässt sich mit dem dualen Baum einer Triangulierung (eines hinreichend großen n-Ecks, in welchem eine Kante ausgezeichnet ist) identifizieren.

(b) Was ist die Binärdarstellung von $(23)_{15}$ (Darstellung zur Basis 15)?

☐ 100001 ☐ 100011 ☐ 110001 ☐ 100010

(c) Welcher der folgenden Ausdrücke liefert in Python 3 `False`?

- ☐ `1.5 == 0.5 + 1.0`
- ☐ `0.5 + 1.0 == 1.0 + 0.5`
- ☐ `0.03 * 0.5 == 0.06`
- ☐ `1.5 == 0.5 + 2 * 0.5`

(d) Wir nennen das *Gerüst* eines Graphen einen aufspannenden Teilgraphen, dessen jede Zusammenhangskomponente ein Baum ist. Es sei nun G ein ungerichteter Graph mit drei Zusammenhangskomponenten, G_1, G_2, G_3, welche jeweils n Kanten haben, wobei $n \geq 3$. Es sei dabei G_1 minimal zusammenhängend, G_2 habe die Knotenmenge $\{0, 1, \ldots, n\}$ und die Kantenmege

$$\{\{0,1\},\{1,2\},\ldots,\{n-1,n\}\}$$

und G_3 sei maximal kreisfrei. Wie viele Gerüste mit exakt drei Zusammenhangskomponenten hat G?

☐ $2n$ ☐ n ☐ 1 ☐ $n-1$

(e) Seien $f, g, h\colon \mathbb{N} \to \mathbb{N}$. Welche der folgenden Aussagen ist **wahr**?

- ☐ Ist $f \in \mathcal{O}(g)$ und $g \in \mathcal{O}(h)$, so ist $h \in \Omega(f)$.
- ☐ Ist $f \in \Omega(g)$, so ist $f \in \mathcal{O}(g^2)$.
- ☐ Ist $f \in \Omega(g)$, so gibt es ein $\alpha > 1$ und ein $N > 100$, so dass für alle $n \in \mathbb{N}$ mit $n \geq N$ gilt, dass $f(n) \geq \alpha g(n)$.
- ☐ Ist $f \in \mathcal{O}(g)$, so gibt es für alle $N > 100$ ein $n \geq N$, so dass für alle $\alpha > 1$ die Ungleichung $f(n) \leq \alpha g(n)$ gilt.

8. Aufgabe

(8 Punkte)

In dieser Aufgabe soll für eine Menge von Tripeln untersucht werden, zwischen welchen Positionen ein funktionaler Zusammenhang besteht.

Sei $M = \{(a_{0,0}, a_{0,1}, a_{0,2}), (a_{1,0}, a_{1,1}, a_{1,2}), \ldots, (a_{n-1,0}, a_{n-1,1}, a_{n-1,2})\} \subseteq \mathbb{N}^3$ eine Menge von Tripeln natürlicher Zahlen. Für $i, j \in \{0, 1, 2\}$ mit $i \neq j$ sagen wir, dass ein *funktionaler Zusammenhang* von Position i zu Position j besteht, falls für alle $k, \ell \in \{0, \ldots, n-1\}$ mit $a_{k,i} = a_{\ell,i}$ auch $a_{k,j} = a_{\ell,j}$ gilt.

Beispiel: In $M = \{(0,0,0), (0,1,0), (2,3,1), (4,2,1)\}$ besteht ein funktionaler Zusammenhang von Position 0 zu Position 2, von Position 1 zu Position 0 sowie von Position 1 zu Position 2. Es besteht **kein** funktionaler Zusammenhang von Position 0 zu Position 1, von Position 2 zu Position 1 sowie von Position 2 zu Position 0.

Schreiben Sie eine `python`-Funktion, die eine Liste der Paare von Positionen berechnet, zwischen denen ein funktionaler Zusammenhang besteht. Sie dürfen alle `python`-Befehle benutzen, die ohne das Importieren zusätzlicher Pakete in `python3` zur Verfügung stehen.

Aufrufparameter Liste

$$L = [(a_{0,0}, a_{0,1}, a_{0,2}), (a_{1,0}, a_{1,1}, a_{1,2}), \ldots, (a_{n-1,0}, a_{n-1,1}, a_{n-1,2})]$$

von Tripeln nicht negativer ganzer Zahlen. Dabei ist die Länge n der Liste eine beliebige nicht negative ganze Zahl.

Rückgabewert Liste mit allen Paaren (i, j) mit $i, j \in \{0, 1, 2\}$, $i \neq j$, für die ein funktionaler Zusammenhang von Position i zu Position j besteht.