Technische Universität Berlin
Fakultät II, Institut für Mathematik
Sekretariat MA 5–2, Dorothea Kiefer
Prof. Dr. Martin Skutella
Georg Loho, Dr. Frank Lutz, Dr. Torsten Mütze

SS 2016

Modulklausur zur Computerorientierten Mathematik I+II

23.07.2016, Beginn: 15:15 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: **28**) **Punkten** bestanden.*

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

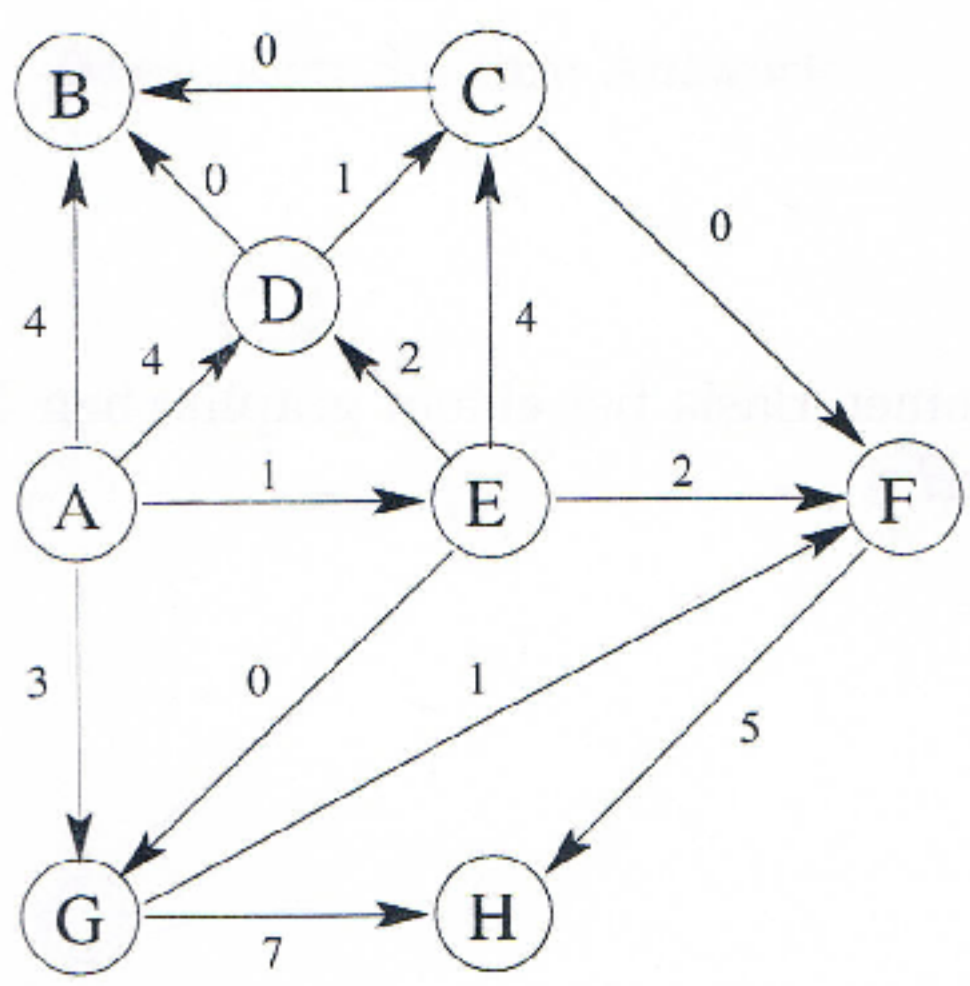
1. Aufgabe (2 + 4 Punkte)

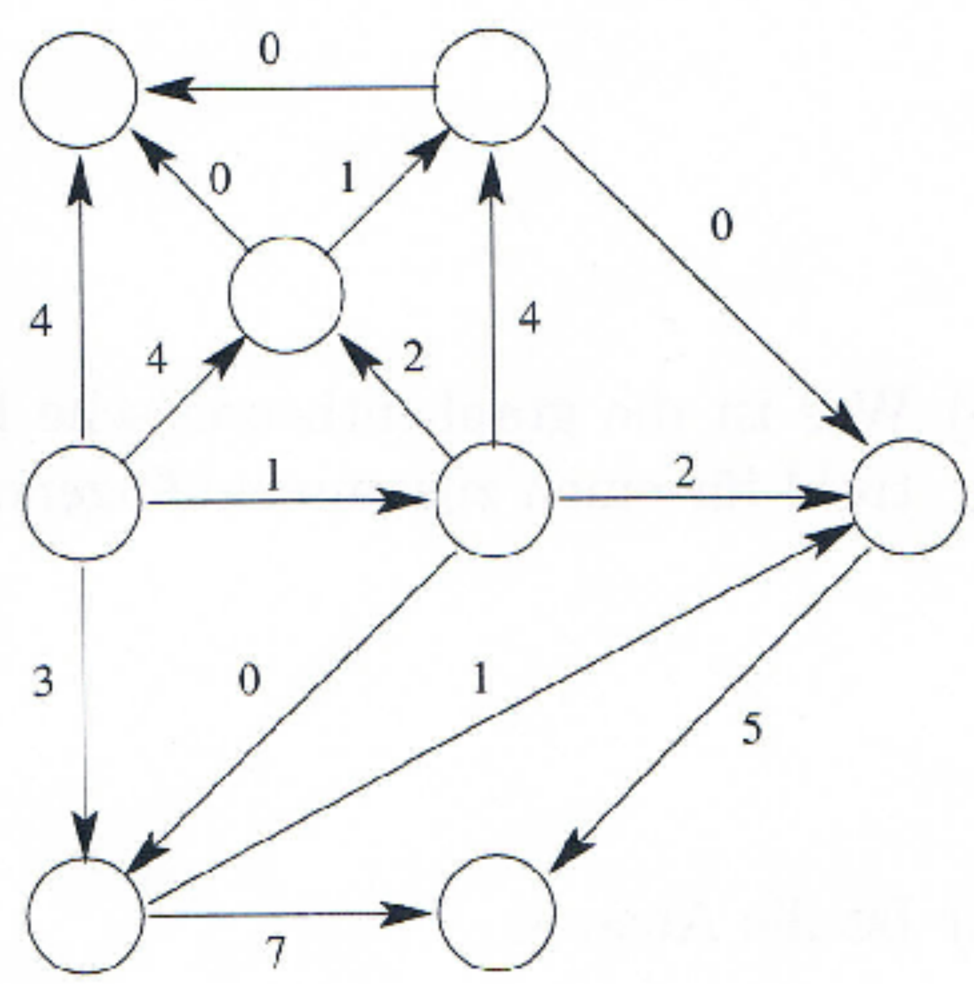
Sei $G = (V, E)$ ein gerichteter Graph mit Kantengewichten $w_e \in \mathbb{Z}$, $e \in E$.

(a) Beschreiben Sie für G und einen Startknoten $r \in V$ als Input die Ausgabe des Bellman-Ford-Algorithmus angewendet auf G.

Welche Worst-Case-Laufzeit (in Θ-Notation) hat der Bellman-Ford-Algorithmus?

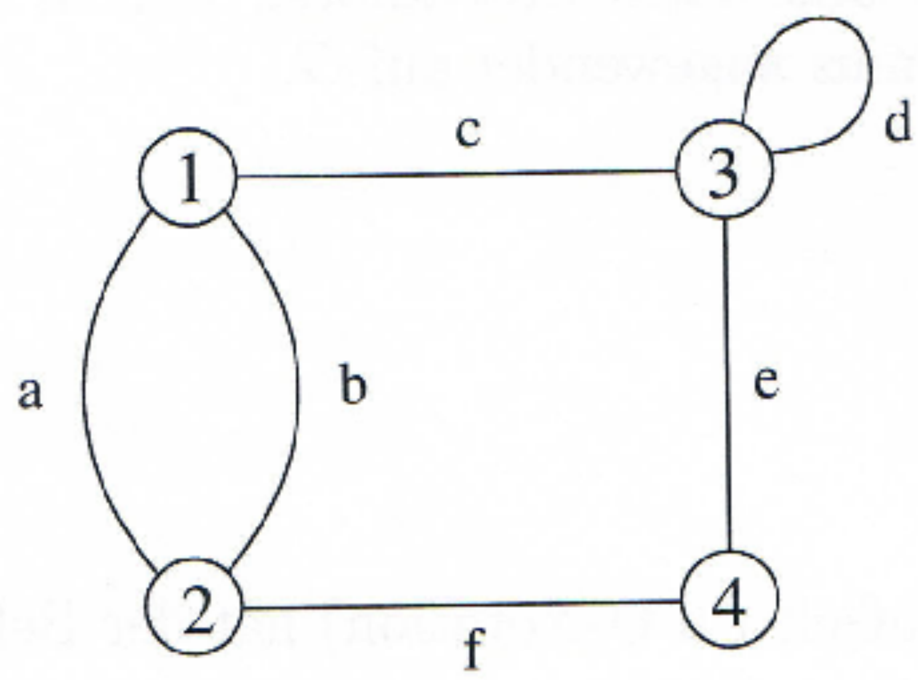
(b) Geben Sie für den folgenden Graphen (links) jeweils explizit einen kürzesten Weg und dessen Länge vom Knoten A zu allen anderen Knoten an. Schreiben Sie in die Knoten der Kopie des Graphen (rechts) die Reihenfolge, in der die Knoten im Durchlauf von Dijkstras Algorithmus aus der Prioritätswarteschlange extrahiert werden.





2. Aufgabe (2 + 1 + 2 + 1 + 2 Punkte)

Sei $G = (V, E)$ der folgende ungerichtete Graph mit Knotenmenge $V = \{1, 2, 3, 4\}$ und Kantenmenge $E = \{a, b, c, d, e, f\}$:



(a) Geben Sie für das graphische Matroid $\mathcal{M} = (E, \mathcal{I})$ zum Graphen G die Menge $\mathcal{B}$ der Basen von $\mathcal{M}$ an.

(b) Was ist die graphentheoretische Bedeutung einer Basis bei einem graphischen Matroid für einen zusammenhängenden Graphen?

(c) Ist die Aussage

„Alle minimal abhängigen Teilmengen von E (aus (a)) haben dieselbe Kardinalität“

wahr oder falsch?

Begründen Sie Ihre Antwort:

(d) Begründen Sie, dass für $E' = \{a, b, c, d\}$ und

$$\mathcal{I}' = \{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a,b\}, \{a,c\}, \{b,c\}, \{b,d\}, \{c,d\}, \{a,b,c\}\}.$$

das Paar $(E', \mathcal{I}')$ ein Unabhängigkeitssystem bildet.

(e) Bildet darüberhinaus das Paar $(E', \mathcal{I}')$ aus Teil (d) ein Matroid?

Begründen Sie Ihre Antwort:

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

(a) Tragen Sie in die folgende Tabelle für die entsprechenden Sortierverfahren die Anzahl der Zuweisungen und Vergleiche auf einem Array der Länge n im Worst-Case in Θ-Notation ein. (Hinweis: Unter einer Zuweisung verstehen wir das Abspeichern eines Elements an einem Speicherplatz.)

Verfahren	QuickSort	MergeSort	InsertionSort (bin. Suche)
Vergleiche			
Zuweisungen			

(b) Formulieren Sie die untere Laufzeitschranke für vergleichsbasiertes Sortieren als Theorem.

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

(d) Beschreiben Sie kurz in Textform einen möglichst effizienten Algorithmus, der überprüft, ob ein Array paarweise verschiedener natürlicher Zahlen der Größe nach sortiert ist. (Hinweis: Beachten Sie, dass eine Sortierung nicht notwendigerweise aufsteigend sein muss.)

(e) Führen Sie MergeSort (rekursive Version) für das folgende Array durch:

$$[8, 6, 1, 3, 7, 4, 5, 2].$$

Geben Sie hierbei die Zwischenergebnisse an und achten Sie insbesondere darauf, dass die Abarbeitungsreihenfolge der Operationen ersichtlich ist.

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

(a) Erklären Sie die Begriffe Präfixcode und Blockcode. Wann ist ein Code eindeutig dekodierbar? Sind Präfixcodes und Blockcodes eindeutig dekodierbar?

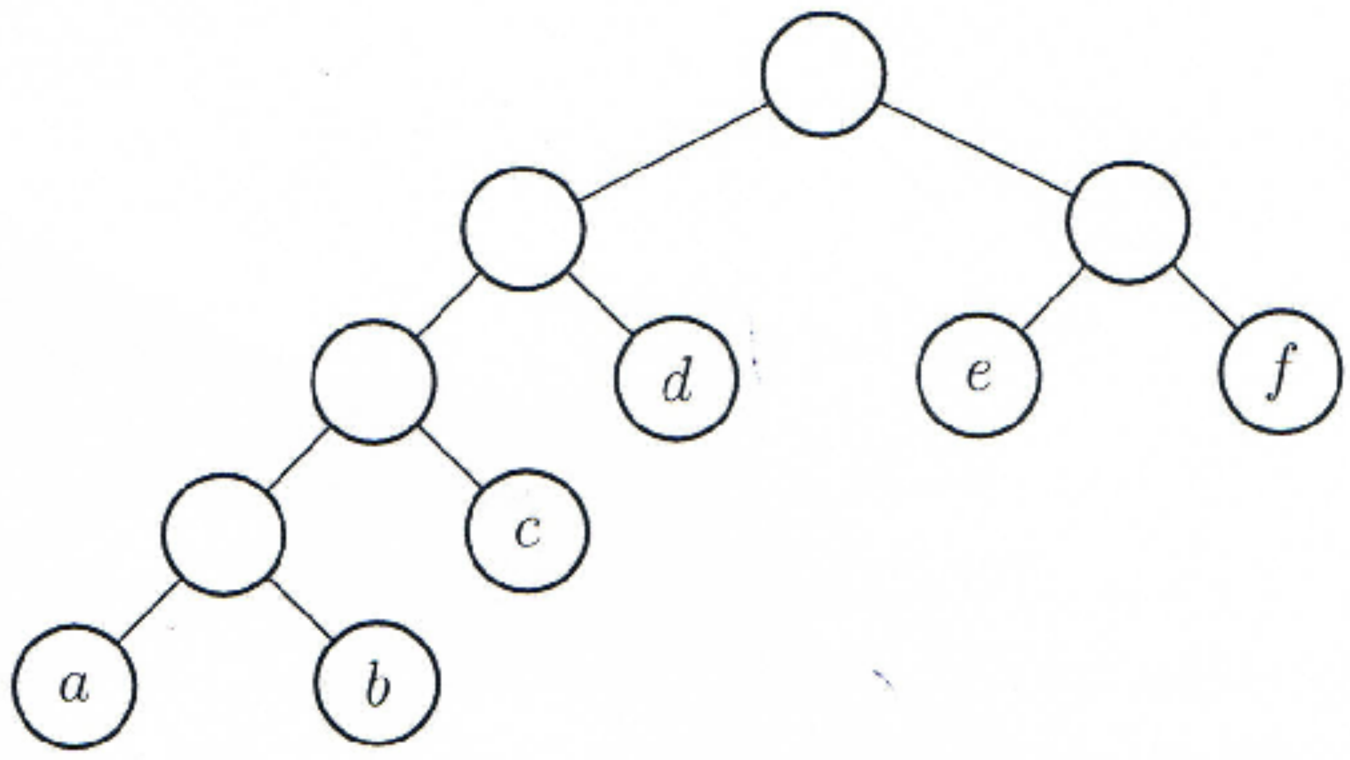
(b) Ihnen ist die folgende Codetabelle gegeben.

a	b	c	d	e
000	010	1	001	011

Zeichnen Sie den zugehörigen Huffmanbaum.

(c) Wann ist ein Präfixcode auf den Zeichen $c_1, \ldots, c_n$ mit zugehörigen Häufigkeiten $f(c_1), \ldots, f(c_n)$ nach Definition optimal?

(d) Ihnen ist der folgende Huffmanbaum mit zugehörigen Zeichen und Häufigkeiten gegeben. Bestimmen Sie alle möglichen Werte für die Häufigkeit von d, sodass der Huffmanalgorithmus den vorliegenden Baum berechnen kann. Begründen Sie Ihre Antwort.



a	b	c	d	e	f
3	3	5	?	12	15

5. Aufgabe (2 + 3 + 4 Punkte)

(a) Definieren Sie die Begriffe Suchbaum und AVL-Baum. Hierbei können die Definition eines Binärbaums und damit verbundene Begrifflichkeiten als bekannt vorausgesetzt werden.

(b) Fügen Sie in einen anfänglich leeren AVL-Baum sukzessive die Schlüssel $1, 5, 2, 4, 3$ ein. Zeichnen Sie den Baum nach jedem Einfügen und geben Sie gegebenenfalls die zugehörige Rotation an.

(c) Geben Sie an, wie man die Anzahl der verschiedenen Suchbäume auf der Schlüsselmenge $1, \ldots, n$ rekursiv bestimmen kann. Begründen Sie ihre Antwort.

6. Aufgabe (2 + 3 Punkte)

(a) Erläutern Sie das Prinzip der optimalen Substruktur an einem Beispiel Ihrer Wahl und geben Sie ein weiteres Beispiel an, wo dieses Prinzip verwendet wird.

(b) In dieser Aufgabe sollen Sie einen optimalen statischen Suchbaum berechnen, wobei nur der letzte Schritt des dynamischen Programms auszuführen ist. Die Schlüssel s_i mit ihren Zugriffswahrscheinlichkeiten β_i sind in der folgenden Tabelle gegeben:

s_i	1	2	3	4	5
β_i	0.5	0.2	0.15	0.05	0.1

Die mittleren Zugriffszeiten für Teilbäume mit höchstens 4 Knoten seien bereits bekannt. In der folgenden Tabelle steht für $i < j$ im (i, j)-ten Eintrag ein Paar (c, k). Hierbei ist c die mittlere Zugriffszeit für einen optimalen statischen Suchbaum auf der Schlüsselmenge $i + 1, \ldots, j$ und k ein Index, für den das Minimum angenommen wurde.

	0	1	2	3	4	5
0	–	$(0.5, 1)$	$(0.9, 1)$	$(1.35, 1)$	$(1.55, 1)$	?
1	–	–	$(0.2, 2)$	$(0.5, 2)$	$(0.65, 2)$	$(0.9, 3)$
2	–	–	–	$(0.15, 3)$	$(0.25, 3)$	$(0.5, 3)$
3	–	–	–	–	$(0.05, 4)$	$(0.2, 5)$
4	–	–	–	–	–	$(0.1, 5)$

Bestimmen Sie nun die mittlere Zugriffszeit für einen optimalen statischen Suchbaum auf der Schlüsselmenge $1, \ldots, 5$ und geben Sie außerdem einen zugehörigen Baum an.

7. Aufgabe (2 + 4 + 2 Punkte)

(a) In eine anfangs leere Hashtabelle der Größe $m = 8$ sollen mittels Divisionsmethode die $n = 6$ Schlüssel 67, 14, 200, 142, 3, 99 eingefügt werden. Wie sieht die resultierende Hashtabelle aus, wenn Kollisionen durch Verkettung aufgelöst werden?

Wie sieht die resultierende Hashtabelle aus, wenn Kollisionen durch lineares Sondieren mit Schrittweite 1 aufgelöst werden?

(b) Berechnen Sie den erwarteten Abstand modulo m von zwei Schlüsseln, die zufällig und gleichverteilt in eine anfangs leere Hashtabelle der Grösse m mit offener Adressierung und linearem Sondieren mit Schrittweite 1 eingefügt werden. Dabei sei $m \geq 2$ eine gerade Zahl. Der Abstand modulo m von Positionen i, j mit $0 \leq i < j \leq m - 1$ ist definiert als $\min(j - i, m - j + i)$.

(c) Beschreiben Sie kurz zwei konkrete Anwendungen/Szenarien, wo Hashing/Hashfunktionen in der Praxis vorkommen oder verwendet werden.

8. Aufgabe (2 + 2 + 2 Punkte)

Gegeben sei die folgende deterministische Turingmaschine T mit Startzustand q_0, Zustandsmenge $Q = \{q_0, q_1, q_2, q_3\}$, Eingabealphabet $\Sigma = \{0, 1\}$, Bandalphabet $\Gamma = \{0, 1, B\}$ und Menge akzeptierender Endzustände $F = \{q_3\}$:

δ	0	1	B
q_0	$(q_0, 0, \mathtt{R})$	$(q_1, 1, \mathtt{R})$	$(q_3, B, \mathtt{N})$
q_1	$(q_1, 0, \mathtt{R})$	$(q_2, 1, \mathtt{R})$	–
q_2	$(q_2, 0, \mathtt{R})$	$(q_0, 1, \mathtt{R})$	–
q_3	–	–	–

Hier bedeutet '–' in Zeile q und Spalte a den Eintrag $(q, a, \mathtt{N})$.

(i) Geben Sie eine informelle Beschreibung der Zustände q_0, q_1 und q_2 an (was 'macht' die Maschine in diesen Zuständen?).

(ii) Welche Sprache $L \subseteq \Sigma^*$ akzeptiert die obige Turingmaschine? Es genügt, die gesuchte Sprache ohne Beweis anzugeben.

(iii) Wie ist T zu modifizieren, so dass die resultierende Turingmaschine T' die Sprache $\Sigma^* \setminus L$ akzeptiert?

9. Aufgabe (7 Punkte)

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

(a) Für die Funktion $f(n) = n(n-1)/2$ gilt

☐ $f(n) \in \Theta(n^2)$ ☐ $f(n) \in \Omega(n^2 \log n)$ ☐ $f(n) \in \mathcal{O}(n)$ ☐ $f(n) \in \Theta(n^{1.5})$

(b) Was ist die Binärdarstellung von $z = (1120)_3$, wobei z in Ternärdarstellung gegeben ist?

☐ $(111111)_2$ ☐ $(10101)_2$ ☐ $(101010)_2$ ☐ $(101101)_2$

(c) Für einen gegebenen Maschinenzahlbereich $F(2, 3, -4, 5)$ sei $\odot$ die Maschinenzahlmultiplikation. Weiter seien die folgenden zwei Aussagen gegeben:

A: „$\odot$ ist assoziativ." B: „$\odot$ ist kommutativ."

Dann gilt

☐ $A \wedge B$ ☐ $A \wedge \neg B$ ☐ $\neg A \wedge B$ ☐ $\neg A \wedge \neg B$

(d) Auf jedem Graphen mit n Knoten und positiven Kantengewichten benötigt der Algorithmus von Dijkstra zur Bestimmung eines Kürzeste-Wege-Baums mit gegebener Wurzel eine Worst-Case-Laufzeit von

☐ $\Omega(n^2)$ ☐ $\mathcal{O}(n^2)$ ☐ $\mathcal{O}(n \log n)$ ☐ $\Theta(n^3)$

(e) Werden n Schlüssel zufällig und gleichverteilt auf eine Hashtabelle der Größe m mit Verkettung gehasht, so hat jeder Tabelleneintrag im Erwartungswert eine Länge von

☐ $\Theta(n/m^2)$ ☐ $\Theta(n^2/m)$ ☐ $\Theta(n \cdot m)$ ☐ $\Theta(n/m)$

(f) Sei HEX(n,m) ein unsymmetrisches HEX-Brett mit $n \neq m$ und HEX(n) ein symmetrisches HEX-Brett (wobei *ohne* Rollentauschmöglichkeit nach dem ersten Zug gespielt wird und $n, m \geq 2$ sei). Welche der folgenden Aussagen ist wahr?

☐ Bei HEX(n,m) hat für alle $n \neq m$ die erste Spielerin eine Gewinnstrategie.
☐ Bei HEX(n,m) kann, falls $n, m \geq 3$ und $n \neq m$, die Spielerin mit der längeren Seite (aus Nettigkeit) die Spielerin mit der kürzeren Seite gewinnen lassen.
☐ Bei HEX(n) hat die zweite Spielerin eine Gewinnstrategie.
☐ Bei HEX(n) kann es ein Unentschieden geben.

(g) Welche der folgende Aussagen ist richtig unter der Annahme $P \neq NP$?

☐ Das Problem „Besitzt eine gegebene aussagenlogische Formel in konjunktiver Normalform eine erfüllende Variablenbelegung?" ist in P.
☐ Es gilt $P \supsetneq NP$.
☐ Das Problem „Sind zwei Knoten in einem gegebenen Graphen durch einen Weg verbunden?" ist in NP.
☐ Das Problem „Gegeben ein Graph G und eine natürliche Zahl k, besitzt G eine Clique (= vollständiger Teilgraph) der Größe k?" ist in P.

10. Aufgabe (7 + 2 + 2 Punkte)

(a) Schreiben Sie einen Algorithmus in Python oder Pseudo-Code, der nachfolgendes Problem löst. (Sie dürfen dabei keine nicht-trivialen Codestücke als gegeben voraussetzen, insbesondere dürfen Sie `sort` und `sorted` nicht verwenden.)

Als Input ist eine Liste $L = [a_1, \ldots, a_{2k+1}]$ von paarweise verschiedenen positiven ganzen Zahlen mit $k \in \mathbb{N}$ gegeben. Es sei m der Median der Liste L. Bestimmen Sie die kleinste Primzahl, die echt größer als der Median m von L ist.

Bemerkung: Eine natürliche Zahl größer 1 heißt *Primzahl*, wenn sie nur durch 1 und sich selbst teilbar ist. Das Bertrandsche Postulat besagt, dass zwischen einer natürlichen Zahl größer 1 und ihrem Doppelten immer mindestens eine Primzahl liegt.

(b) Geben Sie eine möglichst kleine Laufzeitschranke (in $\mathcal{O}$-Notation) für Ihren Algorithmus in Abhängigkeit von k und m an.

(c) Geben Sie einen alternativen (Teil-)Ansatz zur algorithmischen Lösung des obigen Problems an (kurz in Textform, kein Code!) und vergleichen Sie die Ansätze.