Technische Universität Berlin **SS 2024**
Fakultät II, Institut für Mathematik
Sekretariat MA 5–2, Dorothea Kiefer-Hoeft
Prof. Dr. Max Klimm
Dr. Tobias Hofmann, Martin Knaack, Marcel Wack

Modulklausur „Computerorientierte Mathematik I+II “

23.09.2024, 15:00 Uhr bis 17:30 Uhr (Bearbeitungszeit: 150 Minuten)

Nachname, Vorname: ____________________

Matrikelnummer: ____________________

Studiengang: ____________________

Füllen Sie bitte zuerst das Deckblatt vollständig und leserlich aus. Damit erklären Sie, dass

- Ihnen die für diese Prüfung relevanten Zulassungsvoraussetzungen aus der StuPO bekannt sind. Ihnen ist außerdem bewusst, dass deren Nichterfüllung zur Ungültigkeit der Prüfung führen kann (§39 Abs. 2 Satz 4 AllgStuPO);
- Ihnen bekannt ist, dass die Teilnahme an der Prüfung eine ordnungsgemäße Anmeldung voraussetzt, andernfalls die Prüfung nicht gültig ist (§39 Abs. 2 AllgStuPO);
- Ihnen bekannt ist, dass eine Prüfung, die unter bekannten und bewusst in Kauf genommenen (sonstigen) gesundheitlichen Beeinträchtigungen abgelegt wird, grundsätzlich Gültigkeit hat.

Für diese Klausur sind keine Hilfsmittel zugelassen. Mit Bleistift oder Rotstift geschriebene Klausuren können nicht gewertet werden.

Zum Bestehen der Klausur sind mindestens **40** (CES/NidI: **34**) **Punkte** zu erzielen.

1	2	3	4	5	6	7	8	$\sum$
10	10	10	10	10	10	10	10	80

1. Aufgabe (10 Punkte)

Kreuzen Sie jeweils die richtige Lösung an. **Pro Teilaufgabe ist genau eine Antwort richtig.** Das Ankreuzen von mehreren Antworten in einer Teilaufgabe wird mit 0 Punkten bewertet.

(i) Welche der folgenden Zahlen in b-adischer Darstellung entspricht der Dezimalzahl 30?

☐ $(41)_7$ ☐ $(111)_5$ ☐ $(101)_3$ ☐ $(11110)_2$

(ii) Welche der folgenden b-Komplementdarstellungen entspricht der Dezimalzahl -15?

☐ 120 mit $b = 3$ ☐ 2102 mit $b = 3$ ☐ 0001 mit $b = 2$ ☐ 1000 mit $b = 2$

(iii) Welche der folgenden Aussagen ist wahr?

☐ Die Relation $a > b$ auf $\mathbb{Z}$ ist eine partielle Ordnung.
☐ Die Relation $a > b$ auf $\mathbb{N}$ ist eine partielle Ordnung.
☐ Die Relation $A \subseteq B$ (A Teilmenge von B) auf Mengen ist eine totale Ordnung.
☐ Die Relation $a|b$ (a teilt b) auf $\mathbb{N}_{\geq 1}$ ist keine totale Ordnung.

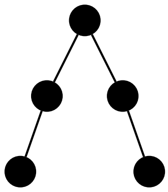
(iv) Welche der folgenden Sortierverfahren hat Worst-Case-Laufzeit $O(n \log n)$ auf einer Eingabe mit n zu sortierenden Elementen?

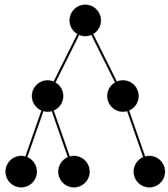
☐ Insertionsort
☐ Mergesort
☐ Quicksort bei Auswahl eines beliebigen Pivotelements
☐ Selectionsort

(v) Welches der folgenden Arrays erfüllt die MaxHeap-Eigenschaft?

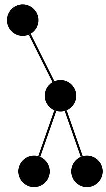
☐ $[7 \quad 3 \quad 6 \quad 2 \quad 4 \quad 5 \quad 1]$
☐ $[7 \quad 3 \quad 6 \quad 2 \quad 1 \quad 4 \quad 5]$
☐ $[1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7]$
☐ $[1 \quad 2 \quad 5 \quad 4 \quad 3 \quad 7 \quad 6]$

(vi) Welcher der folgenden Suchbäume ist kein AVL-Baum?

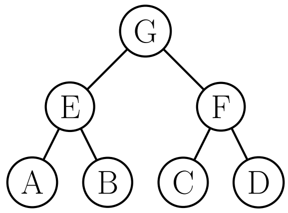
☐ 

☐ 

☐ 

☐ 

(vii) Welche der folgenden ist die Inorder-Reihenfolge der Knoten des angegebenen Binärbaums?



- ☐ GEABFCD
- ☐ AEBGCFD
- ☐ ABCDEFG
- ☐ GEFABCD

(viii) Gegeben sei ein ungerichteter Graph G. Welche der folgenden Aussagen ist nicht äquivalent zu den anderen?

- ☐ G ist kreisfrei und hat $|V(G)| - 1$ Kanten.
- ☐ G ist zusammenhängend und hat $|V(G)| - 1$ Kanten.
- ☐ G ist minimal kreisfrei.
- ☐ G ist minimal zusammenhängend.

(ix) Gegeben sei ein Graph $G = (V, E)$. Welche der folgenden Aussagen ist wahr?

- ☐ Der Algorithmus von Kruskal läuft mit Union-Find Datenstruktur in $O(|E| + |V| \log |V|)$.
- ☐ Der Algorithmus von Bellman-Ford läuft in $O(\log(|V|) \cdot |E|)$.
- ☐ Fords Algorithmus terminiert immer.
- ☐ Das Kürzeste-Wege-Problem für azyklischen Digraphen kann nicht in linearer Zeit $O(|V| + |E|)$ gelöst werden.

(x) Gegeben seien zwei Sprachen $L, M \subseteq \Sigma^*$. Welche der folgenden Aussagen ist falsch?

- ☐ L entscheidbar $\Rightarrow$ M mit $M \subseteq L$ entscheidbar
- ☐ L ist rekursiv $\Rightarrow$ L ist rekursiv aufzählbar.
- ☐ L, M entscheidbar $\Rightarrow$ $L \cup M$ entscheidbar.
- ☐ Ist L NP-vollständig und $L \in P$, so wäre auch $P = NP$.

2. Aufgabe (4 + 2 + 4 Punkte)

(a) Gegeben sei die Matrix

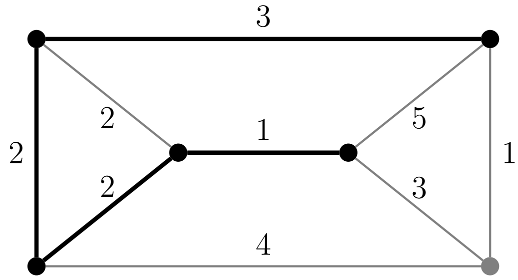
$$I = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}.$$

(i) Zeichnen Sie einen einfachen ungerichteten Graphen mit Inzidenzmatrix I.

(ii) Geben Sie eine Matrix mit Einträgen aus $\{0, 1\}$ an, die nicht die Inzidenzmatrix eines Graphen ist.

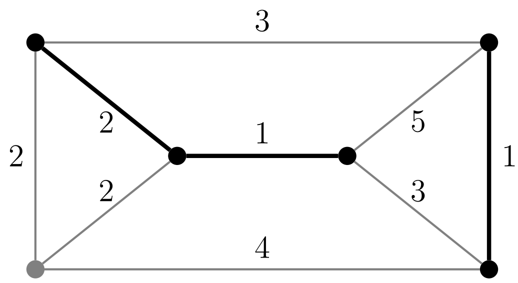
(b) Gibt es einen 3-regulären Graphen mit 7 Knoten? Zeichnen Sie einen solchen oder begründen Sie, warum es einen solchen nicht geben kann.

(c) In den folgenden zwei Abbildungen ist jeweils ein Teil der Kantenmenge eines ungerichteten Graphen G markiert. Kann die jeweilige Kantenmenge ein Zwischenergebnis des Algorithmus von Prim oder Kruskal zur Bestimmung eines minimalen Spannbaums auf G sein? Begründen Sie jeweils kurz ihre Antwort.



Prim:

Kruskal:



Prim:

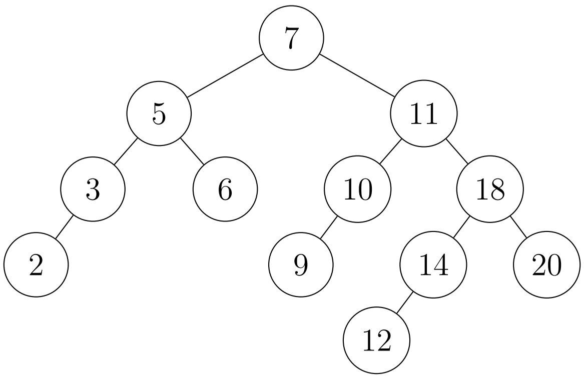
Kruskal:

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

(a) Wann heißt ein Sortierverfahren stabil?

(b) Wann spricht man davon, dass ein Sortierverfahren in-place (oder in situ) arbeitet?

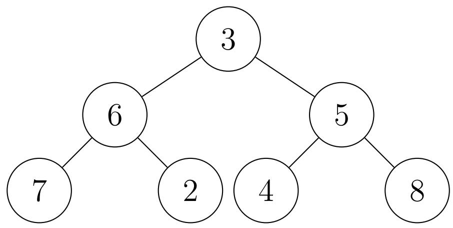
(c) Löschen Sie im folgenden Suchbaum den Schlüssel 2 und überführen Sie den Baum durch geeignete Rotation(en) in einen AVL-Baum.



(d) Füllen Sie die Lücken der folgenden Tabelle aus, um anzugeben, ob die jeweiligen Sortieralgorithmen stabil sind bzw. ob sie in-place arbeiten.

Verfahren	**stabil**	**in-place**
Selection-Sort	**(ja)**	
Merge-Sort		
Quick-Sort		**(ja)**

(e) Wenden Sie den Build-MaxHeap Algorithmus auf den folgenden Binärbaum an. Geben Sie den Baum jedes Mal an, wenn sich die Position von einem Element im Baum verändert.



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

(a) Gegeben sei die Menge $E = \{a, b, c, d\}$. Betrachten Sie zudem zwei Teilmengen $S_1, S_2 \subseteq E$ sowie $I = \{F \subseteq E : |F \cap S_i| \leq 1 \text{ für } i \in \{1, 2\}\}$.

(i) Beweisen Sie, dass (E, I) ein Unabhängigkeitssystem ist.

(ii) Geben Sie passende Mengen S_1 und S_2 an, sodass (E, I) ein Matroid ist. Begründen Sie kurz.

(iii) Gibt es auch S_1 und S_2, sodass (E, I) kein Matroid ist? Falls nein, begründen Sie, warum es solche nicht geben kann. Falls ja, geben Sie passende S_1 und S_2 an und zeigen Sie, dass eines der Matroidaxiome verletzt ist.

(b) Betrachten Sie nun das Matroid (E, I), wobei $E = \{w, x, y, z\}$ die Menge der Spalten der Matrix

$$\begin{array}{c} \begin{array}{cccc} w & x & y & z \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & -1 \\ 2 & 0 & 2 & 0 \\ -1 & 1 & 0 & -1 \end{bmatrix} \end{array}$$

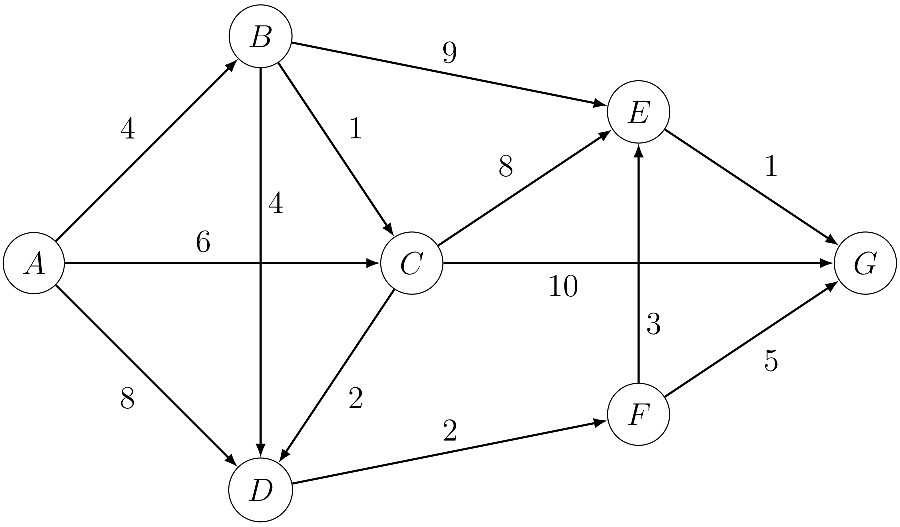
ist und I die Menge der linear unabhängigen Teilmengen von E ist. Geben Sie für $(E, \mathcal{I})$ die Menge aller Basen $\mathcal{B}$ sowie die Menge aller Kreise $\mathcal{C}$ an.

$\mathcal{B} =$

$\mathcal{C} =$

5. Aufgabe (7 + 3 Punkte)

(a) (i) Führen Sie Dijkstras Algorithmus zur Lösung des Kürzeste-Wege-Problems auf dem nachfolgend abgebildeten Graphen mit Startknoten A aus.



Tragen Sie dafür in der folgenden Tabelle in jedem Schritt und für jeden Knoten seine aktuelle Distanz zum Startknoten und den vom Algorithmus bestimmten Vorgängerknoten ein.

Schritt	A	B	C	D	E	F	G
0	0/−	∞/−	∞/−	∞/−	∞/−	∞/−	∞/−
1							
2							
3							
4							
5							
6							

(ii) Geben Sie den kürzesten Pfad vom Startknoten A zum Knoten G an.

(iii) Was ist die Zeitkomplexität von Dijkstra's Algorithmus bei Implementierung mittels MinHeap?

(b) Es seien f und g beliebige Funktionen. Gilt immer, dass $f \in O(g)$ oder $g \in O(f)$? Wenn ja, begründen Sie Ihre Aussage. Wenn nein, geben Sie zwei Funktionen an und begründen Sie kurz, warum weder $f \in O(g)$ noch $g \in O(f)$ gilt.

6. Aufgabe

(1 + 1 + 1 + 2 + 5 Punkte)

(a) Was ist ein Blockcode?

(b) Was ist ein Präfixcode?

(c) Wann heißt ein Präfixcode optimal?

(d) Gibt es eindeutig dekodierbare Codes, die keine Präfixcodes sind? Falls nicht, begründen Sie ihre Antwort. Falls es solche gibt, geben Sie ein Beispiel an.

(e) Gegeben sei die folgende Häufigkeitstabelle für die Buchstaben a, b, c, d, e, f, g, h.

Zeichen	a	b	c	d	e	f	g
Häufigkeit	4	3	5	9	6	2	1

Wenden Sie den Huffman-Algorithmus an und geben Sie den resultierenden optimalen Präfixcode und den zugehörigen binären Baum an.

7. Aufgabe (5 + 2 + 3 Punkte)

Gegeben sei eine deterministische Turingmaschine T durch

- die Zustandsmenge $Q = \{q_0, q_1, q_2, q_3, q_4\}$ mit Startzustand q_0,
- das Eingabealphabet $\Sigma = \{0, 1\}$,
- das Bandalphabet $\Gamma = \Sigma \cup \{X, Y, B\}$,
- die Menge akzeptierender Zustände $F = \{q_4\}$

sowie folgendes Programm:

δ	q_0	q_1	q_2	q_3	q_4
0	$(q_2, X, \mathtt{R})$	$(q_3, Y, \mathtt{L})$	$(q_2, 0, \mathtt{R})$	$(q_3, 0, \mathtt{L})$	—
1	$(q_1, X, \mathtt{R})$	$(q_1, 1, \mathtt{R})$	$(q_3, Y, \mathtt{L})$	$(q_3, 1, \mathtt{L})$	—
X	—	—	—	$(q_0, X, \mathtt{R})$	—
Y	$(q_0, Y, \mathtt{R})$	$(q_1, Y, \mathtt{R})$	$(q_2, Y, \mathtt{R})$	$(q_3, Y, \mathtt{L})$	—
B	$(q_5, B, \mathtt{R})$	—	—	—	—

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

(a) (i) Vervollständigen Sie die folgende Konfigurationsfolge:

$q_0 10011$ $\vdash$ __________ $\vdash$ __________ $\vdash$ __________

$\vdash$ __________ $\vdash$ __________ $\vdash$ __________

$\vdash$ __________ $\vdash$ __________ $\vdash$ __________

(ii) Terminiert die Turingmaschine T in einem akzeptierenden Zustand?

☐ Ja. ☐ Nein.

(b) Kreuzen Sie an, welche der folgenden Eingaben von der Turingmaschine T akzeptiert werden. Beschreiben Sie die Sprache, die T akzeptiert.

☐ 00
☐ 11
☐ 0101010101
☐ 0001111

(c) Beweisen Sie folgende Aussage: Sei Σ ein endliches Alphabet. Wenn $L \subseteq \Sigma^*$ und $\overline{L} := \Sigma^* \setminus L$ semi-entscheidbar (rekursiv aufzählbar) sind, dann ist L entscheidbar (rekursiv).

8. Aufgabe (3 + 3 + 4 Punkte)

Diese Aufgabe besteht aus drei Teilen. In den ersten beiden Teilen sollen Sie die Ausgabe und die Laufzeit von gegebenem Code bestimmen. Im dritten Teil sollen Sie selbst eine Funktion schreiben. Aufgabenteile (a) und (b) sind Aufgaben in `julia`, in Aufgabenteil (c) können Sie in der Sprache Ihrer Wahl lösen.

(a) Betrachte das folgende Codefragment `julia` Gib die Laufzeit der Funktion `f2` für Integervektoren der Länge N in O-Notation an und bestimme die Ausgabe des Codefragments für folgende Eingabe

```julia
f2([3,1,4,1,5,9,2,6,5,3,5])
```

Dabei ist die Funktion `f2` durch den folgenden Code definiert:

```julia
function f1(left::Vector{Int}, right::Vector{Int})::Vector{Int}
  result = Int[]
  sizehint!(result, length(left) + length(right))

  while !isempty(left) && !isempty(right)
    push!(result, left[1] >= right[1] ? popfirst!(left) : popfirst!(right))
  end
  append!(result, left)
  append!(result, right)
  return result
end

function f2(v::Vector{Int})::Vector{Int}
  length(v) <= 1 && return v
  middle = div(length(v),2)

  left = f2(v[1:middle])
  right = f2(v[middle+1:end])
  return f1(left, right)
end
```

Laufzeit: ________

(b) Der folgende `julia` Code transponiert eine Matrix $\mathtt{M} \in \mathbb{Z}^{n \times n}$ und speichert die transponierte Matrix in `T`. Geben Sie die Laufzeit des Algorithmus in O-Notation an. Schreiben Sie anschließend eine optimierte Variante des Algorithmus zur Verbesserung der Laufzeit. Inwieweit ändert sich dadurch die Laufzeit in O-Notation?

Hinweis: Sie können ihre optimierte Lösung entweder in `julia` oder `python` schreiben.

```julia
T = zeros(Int, size(M))
for x in CartesianIndices(M)
  T[reverse(x.I)...] = M[x]
end
```

(c) In diesem Aufgabenteil sollen Sie selbst eine `julia`- oder `python`-Funktion schreiben. Dabei können Sie alle sprachspezifischen Befehle verwenden, die ohne den Import zusätzlicher Pakete zur Verfügung stehen.

Schreiben Sie eine `julia` oder `python` Funktion `intersection(u, v)`, wobei `u` und `v` Vektoren bzw. Listen von Integerzahlen sind. Die Funktion soll die Schnittmenge der beiden Eingaben zurückgeben, d.h. einen Vektor (oder eine Liste) von Zahlen, die in beiden Mengen vorkommen. Die Laufzeit der Funktion soll $O(n+m)$ betragen, wobei n die Länge von `u` und m die Länge von `v` ist. Wähle sie für ihre implementierung eine geeignete Datenstruktur.