Technische Universität Braunschweig
Institut für Betriebssysteme und Rechnerverbund
Abteilung Algorithmik

Prof. Dr. Sándor P. Fekete
Ramin Kosfeld
Chek-Manh Loi

Wintersemester 2023/2024

Klausur
Algorithmen und Datenstrukturen
13.02.2024

Name: .....

Vorname: .....

Matr.-Nr.: .....

Studiengang: .....

☐ Bachelor ☐ Master ☐ Andere

Klausurcode:
Dieser wird benötigt, um das Ergebnis der Klausur abzurufen.

Hinweise:

- Bitte das Deckblatt in Druckschrift vollständig ausfüllen.
- Die Klausur besteht aus 13 Blättern, bitte auf Vollständigkeit überprüfen. Die Heftung darf nicht entfernt werden
- Erlaubte Hilfsmittel: keine
- Eigenes Papier ist nicht erlaubt.
- Die Rückseiten der Blätter dürfen beschrieben werden.
- Die Klausur ist mit 50 % der Punkte bestanden.
- Antworten, die *nicht* gewertet werden sollen, bitte deutlich durchstreichen. Kein Tippex verwenden!
- Mit *Bleistift* oder in *Rot* geschriebene Klausurteile können nicht gewertet werden.
- Werden mehrere Antworten gegeben, werten wir die mit der geringsten Punktzahl.
- Sämtliche Algorithmen, Datenstrukturen, Sätze und Begriffe beziehen sich, sofern nicht explizit anders angegeben, auf die in der Vorlesung vorgestellte Variante.
- Sofern nicht anders angegeben, sind alle Graphen als einfache Graphen zu verstehen.
- Die Bearbeitungszeit für die Klausur beträgt 120 Minuten.

Aufgabe	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	Σ
Punkte	**16**	**12**	**14**	**13**	**10**	**12**	**11**	**12**	**100**
Erreicht									
Note	—	—	—	—	—	—	—	—	

Aufgabe 1: Graphen **(7+3+6 Punkte)**

a) Wende den Algorithmus von Hierholzer auf den Graphen G aus Abbildung 1 an; starte dabei mit dem Knoten v_0. Kommen in einem Schritt des Algorithmus mehrere Knoten in Frage, wähle denjenigen mit dem kleinsten Index. Gib für jede Iteration des Algorithmus den maximal geschlossenen Weg als Knotenliste an und markiere in den Listen, wenn Kreise verschmolzen werden.

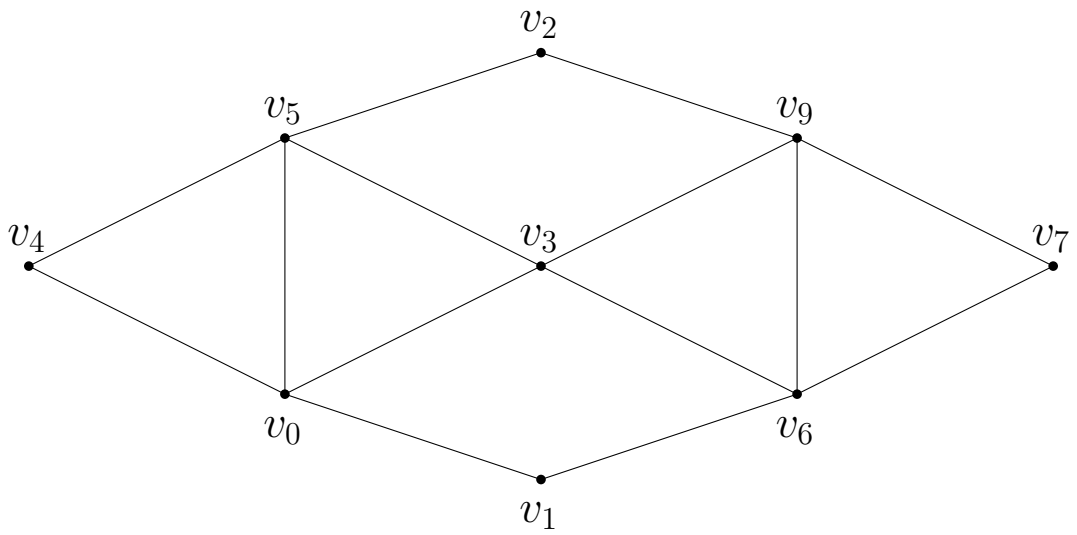


Abbildung 1: Der Graph G

b) Zeige oder widerlege: Jeder zusammenhängende einfache Graph kann mit Kanten (keine Knoten) erweitert werden, sodass der resultierende Graph einfach ist und eine Eulertour enthält.

c) Gib die Adjazenzliste und Inzidenzmatrix des Graphen G' aus Abbildung 2 an. Sortiere alle Einträge in der Liste *und* in der Matrix aufsteigend nach Knoten- und Kantenindizes. Du kannst deine Lösung in den dafür vorgegebenen Bereichen eintragen.

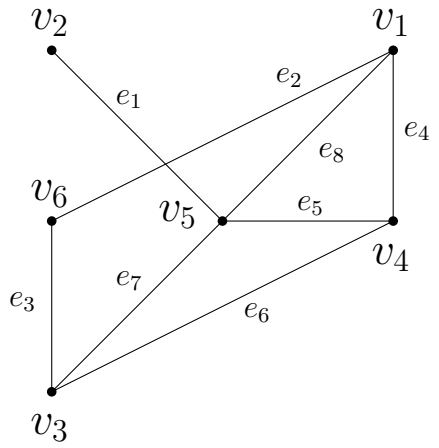


Abbildung 2: Der Graph G'

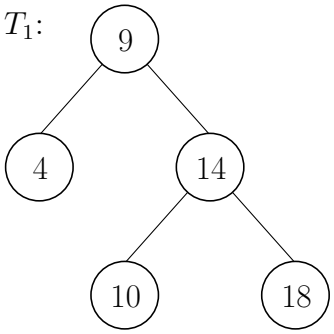
Adjazenzliste:

Inzidenzmatrix:

Aufgabe 2: Dynamische Datenstrukturen **(2+2+2+2+4 Punkte)**

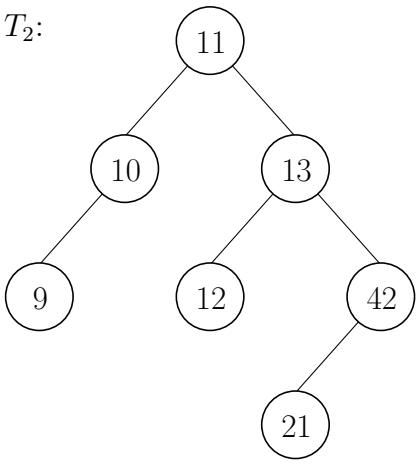
a) Führe INSERT(T_1, 12) auf dem AVL-Baum aus. Gib den Baum nach der Einfügeoperation sowie nach jeder Restructure-Operation an.
(Hinweis: Die Einfügeoperation darf in dem angegebenen Baum durchgeführt werden.)

T_1:



b) Führe DELETE(T_2, 10) auf dem AVL-Baum aus. Gib den Baum nach der Löschoperation sowie nach jeder Restructure-Operation an.

T_2:



c) Wie lange benötigt das Einfügen eines Elementes in eine einfach verkettete Liste? Begründe deine Antwort.

d) Wie lange benötigt das Löschen eines identifizierten Elementes aus einer einfach verketteten Liste? Begründe deine Antwort.

e) AVL-Bäume funktionieren auch, wenn mehrfach derselbe Schlüssel im Baum enthalten ist. Beschreibe, wie für einen gegebenen AVL-Baum B in $\mathcal{O}(\log n)$ ermittelt werden kann, ob alle Schlüssel in B gleich sind.

Aufgabe 3: Wachstum von Funktionen **(4+4+6 Punkte)**

a) Seien $f, g, h : \mathbb{N} \to \mathbb{R}$ Funktionen.
Zeige oder widerlege: $f(n) \in \mathcal{O}(g(n))$ und $g(n) \in \mathcal{O}(h(n)) \Rightarrow f(n) \in \mathcal{O}(h(n))$

b) Bestimme geeignete Konstanten, um zu zeigen, dass folgende Aussage korrekt ist.

$$f(n) \coloneqq 3n^2 - 8n + \log_2 n \in \Omega(n^2)$$

c) In welcher Beziehung stehen die folgenden Klassen zueinander? Schreibe $\subsetneq$ in das Feld, wenn Klasse A in Klasse B enthalten, aber nicht gleich ist, $\supsetneq$, wenn Klasse B in Klasse A enthalten, aber nicht gleich ist, $=$, wenn die Klassen A und B übereinstimmen und $\times$, wenn dies alles nicht zutrifft. Eine Begründung ist nicht notwendig.

A	Relation	B
$\mathcal{O}(n^2)$		$\mathcal{O}(2^n)$
$\Theta(n^2)$		$\mathcal{O}(n^3)$
$\mathcal{O}\left(\sum_{i=1}^{n} i\right)$		$\mathcal{O}(n^3)$
$\Omega(n)$		$\Theta\left(\frac{n}{\log n}\right)$
$\Omega(\log n)$		$\mathcal{O}(\log(\log n))$
$\Omega\left(\sin(n) + 2\right)$		$\Omega\left(1\right)$

Aufgabe 4: Rekursionen **(13 Punkte)**

Bestimme das asymptotische Wachstum der folgenden Funktionen mithilfe des in der Vorlesung vorgestellten Mastertheorems, oder begründe in einem Satz, warum man das Theorem nicht anwenden kann. Gib beim Anwenden alle im Mastertheorem auftretenden Parameter an.

a) $R(n) = 4 \cdot R\left(\frac{n}{9}\right) + 2n - R\left(\frac{n}{3}\right)$

b) $T(n) = 8n + 64 \cdot T\left(\frac{n}{4}\right) - 21 \log n - 6 + n^2$

c) $U(n) = 3 \cdot U\left(\frac{n}{3}\right) + 8\sqrt{n} + 53 \cdot U\left(\frac{n}{9}\right) + 4n^2$

d) $S(n) = n + n \cdot S(\frac{n}{4})$

e) $V(n) = 3 \cdot V\left(\frac{n}{9}\right) + 7$

Aufgabe 5: Mediane **(2+2+6 Punkte)**

a) Sei X eine Menge von paarweise verschiedenen natürlichen Zahlen. Wie lautet die Definition eines Rang-k Elements in X?

b) Betrachte die Menge $X = \{100, 0, 42, 29, 9, 2\}$. Bestimme folgende Elemente.

Rang-5 Element: _____

Alle Mediane: _____

c) Beschreibe die Schritte, die benötigt werden, um das Rang-k Element in einer Menge X von paarweise verschiedenen Zahlen in $O(n)$ Zeit zu finden.
(Hinweis: Es sind kein Korrektheitsbeweis, Pseudocode oder Laufzeitanalyse erforderlich.)

Aufgabe 6: Sortieren **(6+3+3 Punkte)**

a) Sortiere das Array aus Abbildung 3 mit dem QuickSort-Algorithmus aus der Vorlesung. Gib das Ergebnis von jedem Partition-Aufruf an, sofern sich das Array ändert. Markiere außerdem das genutzte Pivot-Element.
(Hinweis: Der Platz unter der Tabelle kann für Nebenrechnungen verwendet werden.)

$A =$	1	3	4	5	9	8	2	7	6
1. $A =$									
2. $A =$									
3. $A =$									
4. $A =$									
5. $A =$									

Abbildung 3: QuickSort auf Array A.

b) Welche Eigenschaft erfüllt ein stabiler Sortieralgorithmus? Ist CountingSort in der aus der Vorlesung bekannten Version stabil? Begründe deine Antwort.

c) Welche Laufzeit besitzt QuickSort für n Zahlen im Worst-Case? Begründe kurz, unter welchen Voraussetzungen dieser schlechteste Fall eintritt.

Aufgabe 7: Algorithmenentwurf **(7+4 Punkte)**

a) Sei B ein binärer Suchbaum mit Wurzel r. Wir wollen die Anzahl der Knoten in B bestimmen, die einen geraden Schlüssel haben.

Wir wollen zeigen, dass das Problem in $O(n)$ Zeit lösbar ist. Gib dazu einen geeigneten **rekursiven** Algorithmus (CountEven(v)) in Pseudocode mit maximal 10 Zeilen an und begründe, dass die Laufzeit eingehalten wird. Ein Korrektheitsbeweis ist nicht erforderlich.

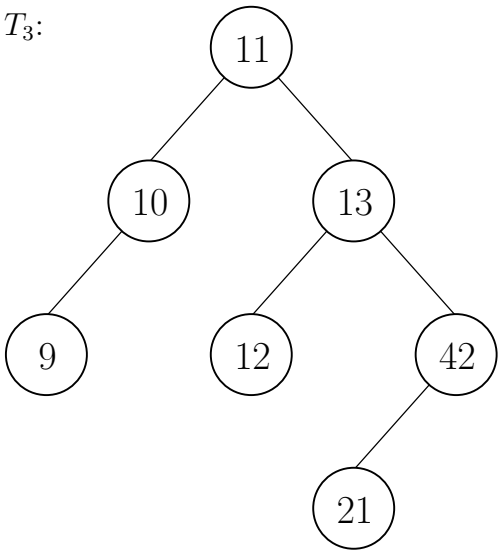
(Hinweis: Es darf angenommen werden, dass der Algorithmus mit einem Aufruf von CountEven(r) gestartet wird.)

b) Algorithmus 1 gibt einen binären Baum in postorder aus. Wende den Algorithmus auf den binären Baum T_3 an. Gib dabei die Reihenfolge an, in der die Knoten ausgegeben werden, wenn PostOrder mit dem Wurzelknoten von T_3 ausgeführt wird.

```
function POSTORDER(v)
    if v ≠ Nil then
        POSTORDER(l[v])
        POSTORDER(r[v])
        Gib S[v] aus
```

Algorithmus 1: Postorder Ausgabe eines binären Baumes.

T_3:



Aufgabe 8: Kurzfragen **(2+2+2+2+2+2 Punkte)**

Kreuze die korrekten Aussagen an. Es gibt nur Punkte für vollständig korrekt angekreuzte Teilaufgaben.

(Hinweis: In jeder Teilaufgabe ist immer mindestens eine Aussage korrekt.)

a) In einem einfachen Graphen ...

- ... ist die Anzahl der Knoten mit ungeradem Grad immer gerade. ☐
- ... gibt es weder doppelte Kanten noch Schleifen. ☐
- ... gibt es maximal so viele Kanten, wie der Graph Knoten enthält. ☐

b) In welcher Datenstruktur kann das größte Element in $O(1)$ Zeit gefunden werden?

- AVL-Baum ☐
- Max-Heap ☐
- Stapel ☐

c) Welche Laufzeitschranken sind korrekt?

- Vergleichsbasiertes Sortieren von n Zahlen: $\Omega(n^2)$ ☐
- Finden einer Eulertour in einem Graphen: $O(n + m)$ ☐
- Test auf Zusammenhang eines Graphen: $O(n + m)$ ☐

d) MERGESORT...

- ... basiert auf dem Prinzip „Teile und Herrsche“. ☐
- ... ist ein vergleichsbasiertes Sortierverfahren. ☐
- ... hat im Best-Case eine Laufzeit von $O(n)$. ☐

e) Wird eine Subroutine, welche die Laufzeit $\mathcal{O}(\log n)$ besitzt, $\mathcal{O}(n)$ mal wiederholt, so ist die asymptotische Gesamtlaufzeit der wiederholten Durchführung der Subroutine...

- ... unbekannt, da zunächst die absolute Laufzeit bekannt sein muss. ☐
- ... $\mathcal{O}(n \log n)$. ☐
- ... $\mathcal{O}(n)$. ☐

f) In einem Programm benötigen wir eine Datenstruktur, die ganze Zahlen speichert. Nachdem in diese Struktur einmalig eine große Menge von Zahlen eingefügt wurde, wird die Struktur dauerhaft dafür genutzt, zu prüfen, ob eine bestimmte Zahl in dieser Struktur vorhanden ist. Welche der folgenden Datenstrukturen ist laufzeittechnisch am besten für diese Anwendung geeignet?

- Doppelt verkettete Liste ☐
- AVL-Baum ☐
- Max-Heap ☐

Viel Erfolg ☺