↑ Für deinen Klausurcode

Technische Universität Braunschweig
Institut für Betriebssysteme und Rechnerverbund
Abteilung Algorithmik

Wintersemester 2024/2025

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

Klausur
Algorithmen und Datenstrukturen
12.02.2025

Nachname: ....

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 Bearbeitungszeit für die Klausur beträgt 120 Minuten.

- 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.

Aufgabe	1	2	3	4	5	6	7	8	Σ
Max	12	14	13	13	13	14	9	12	100
Erreicht									

Aufgabe 1: Graphenscan **(7+2+3 Punkte)**

a) Betrachte den Graphen G aus Abbildung 1. Wende Breiten- und Tiefensuche mit Startknoten v_1 auf G an. Falls zu einem Zeitpunkt mehrere Knoten für den nächsten Schritt in Frage kommen, wähle denjenigen mit dem kleinsten Index. Zeichne die resultierenden Bäume in Abbildung 2.

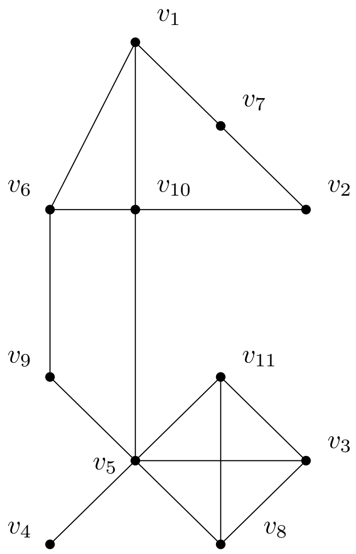
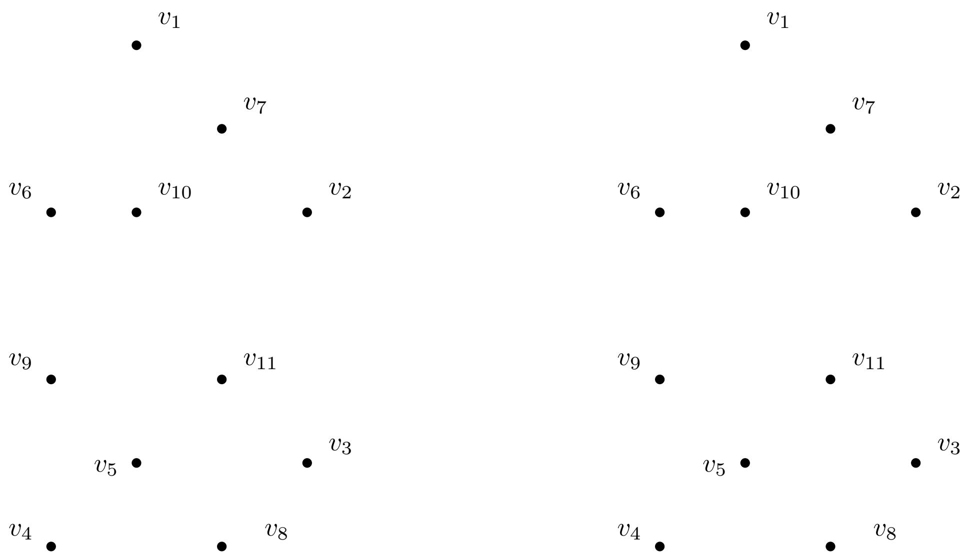


Abbildung 1: Der Graph G.



(a) Breitensuchbaum von G. **(b)** Tiefensuchbaum von G.

Abbildung 2

b) Welche Datenstruktur wird für Tiefensuche benutzt? Welche Laufzeit besitzt Tiefensuche für einen Graphen mit n Knoten und m Kanten?

c) Sei der Graph S aus Abbildung 3 gegeben. Zeige oder widerlege: Es lassen sich Kanten so in S einfügen, dass der resultierende einfache Graph eine Eulertour besitzt.

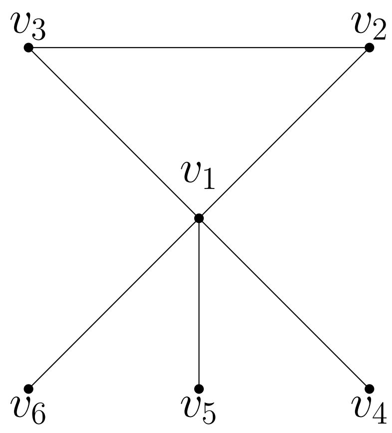
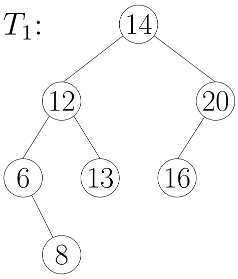


Abbildung 3: Der Graph S.

Aufgabe 2: Dynamische Datenstrukturen **(2+3+2+3+4 Punkte)**

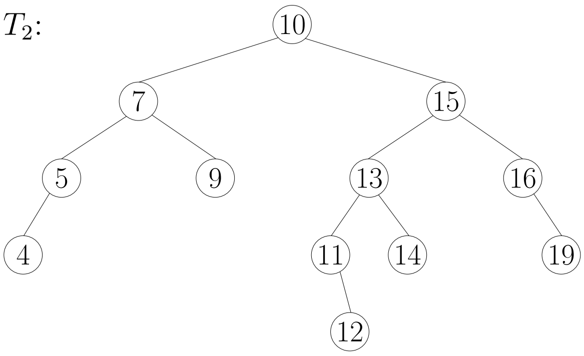
a) Führe INSERT(T_1, 7) auf dem AVL-Baum aus. Gib den Baum nach der Einfügeoperation sowie nach jeder RESTRUCTURE-Operation an.
(Hinweis: Die Einfügeoperation darf in dem angegebenen Baum durchgeführt werden.)

T_1:



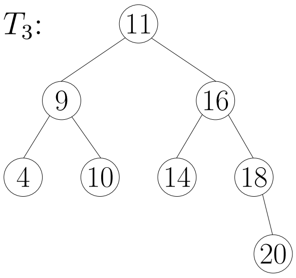
b) Führe DELETE(T_2, 9) auf dem AVL-Baum aus. Gib den Baum nach der Löschoperation sowie nach jeder RESTRUCTURE-Operation an.

T_2:



c) Gib alle Blätter x des AVL-Baums T_3 an, die durch eine DELETE(T_3, x)-Operation entfernt werden können, ohne dass eine Restrukturierung notwendig ist.

T_3:



d) Sei x ein Knoten in einem AVL-Baum T. Beschreibe wie man in $\mathcal{O}(\log n)$ Zeit den Vorgänger von x in T findet. (Hinweis: Hier ist kein Pseudocode nötig.)

e) Beschreibe einen Algorithmus, der eine beliebige Menge M von n Zahlen in einen leeren AVL-Baum einfügt, sodass keine Restrukturierung notwendig ist. (Hinweis: Hier ist kein Pseudocode nötig.)

Aufgabe 3: Wachstum von Funktionen **(5+4+4 Punkte)**

a) Kreuze an, in welchen Klassen die angegebenen Funktionen liegen.

$f(n)$	$\mathcal{O}(n)$	$\Omega(n)$	$\mathcal{O}(n \log n)$	$\Omega(n \log n)$	$\mathcal{O}(n^2)$	$\Omega(n^2)$
$50n + 5n$						
$n^3 - 500n$						
$\log(n^n)$						
3^n						
$\frac{n^2-5}{8n \log n+5}$						

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

$$f(n) := 2n^2 - 7n + 7\log(n) \in \Omega(n^2)$$

c) Seien $f, g : \mathbb{N} \to \mathbb{R}$ Funktionen.
Zeige oder widerlege: $\Omega(f(n)) \cap \mathcal{O}(g(n)) \neq \emptyset \Rightarrow f(n) \in \mathcal{O}(g(n))$

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)

$$T(n) = T\left(\frac{n}{16}\right) + 15n + \sum_{i=1}^{4} T\left(\frac{n}{2^i}\right)$$

b)

$$T(n) = n \cdot T\left(\frac{n}{4}\right) + 3 \cdot T\left(\frac{n}{4}\right)$$

c)

$$T(n) = 64 \cdot T\left(\frac{n}{4}\right) + 15n + \sum_{i=2}^{4} \log(i)$$

d)

$$T(n) = 3n + \frac{1}{2} \cdot T\left(\frac{n}{2}\right)$$

e)

$$T(n) = 14n \log n + 20 \cdot T\left(\frac{n}{5}\right) + 4n^2 + 2 \cdot T\left(\frac{n}{4}\right)$$

Aufgabe 5: Mediane **(1+4+8 Punkte)**

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

b) Wie lautet die Rekursionsgleichung zur Bestimmung der Laufzeit, um ein Rang-k Element zu finden? Beschreibe kurz woher die Teilrekursionen kommen.

c) Brechne das Rang-10 Element der Menge von

$$X = \{14, 9, 5, 10, 4, 3, 6, 7, 2, 15, 8, 13, 12, 11, 1\}.$$

Gib dabei alle Aufrufe des SELECT-Algortihmus und deren Rückgabewerte m an. Gib ausserdem die Fünfergruppen und die Mengen M, X_1, X_2 und X_3 an.

Aufgabe 6: Sortieren **(7+3+4 Punkte)**

a) Sortiere das Array A aus Abbildung 4 mit dem Algorithmus MERGESORT. Gib (separat und in chronologischer Reihenfolge, also **nicht** für mehrere Teilmengen im selben Schritt) das Array A nach jedem Aufruf von MERGE an. Elemente, die nicht zum im jeweiligen Mergeschritt betrachteten Teilarray gehören, müssen nicht angegeben werden. Die Aufrufe von MERGE auf einem Teilarray der Länge 1 müssen nicht angegeben werden. Nutze zur Bearbeitung der Aufgabe die in Abbildung 4 gegebene Tabelle.

A	2	1	5	4	6	7	8	3
1. A								
2. A								
3. A								
4. A								
5. A								
6. A								
7. A								

Abbildung 4: MERGESORT auf dem Array A.

b) QUICKSORT verwendet als Subroutine PARTITION. Wende auf das Array B aus Abbildung 5 PARTITION$(B, 1, 8)$ an. Gib das Ergebnis in der in Abbildung 5 gegebenen Tabelle an. Kennzeichne dabei das Pivotelement und die beiden sich ergebenden Teilarrays. (Hinweis: Es wird immer das letzte Element des Arrays als Pivotelement genutzt.)

B	7	9	2	9	3	12	1	5
1. B								

Abbildung 5: PARTITION auf dem Array B.

c) COUNTINGSORT besitzt eine Laufzeit von $\mathcal{O}(n+k)$. Beschreibe kurz den Algorithmus und gib dabei an, welche Bedeutung die Parameter n und k besitzen.

Aufgabe 7: Algorithmenverständnis **(4+3+2 Punkte)**

Betrachte Algorithmus 1.

Eingabe: Ein Array A mit n Elementen.
Ausgabe: TRUE oder FALSE.

```
function CHECKARRAY(A)
    Stack S                                   ▷ Initialisiere leeren Stack
    for i = 1, ..., length[A] do
        PUSH(S, A[i])
    for i = 1, ..., length[A] do
        if POP(S) ≠ A[i] then
            return FALSE
    return TRUE
```

Algorithmus 1

a) Beschreibe, welche Eigenschaft des Arrays A der Algorithmus 1 überprüft.

b) Welche Laufzeit in Abhängigkeit von der Anzahl der Elemente n hat der Algorithmus? Erkläre kurz, wie diese Laufzeit zustande kommt.

c) Kann es einen korrekten Algorithmus geben, der die Eigenschaft asymptotisch schneller überprüft als Algorithmus 1? Begründe deine Antwort in einem Satz.

Aufgabe 8: Kurzfragen **(2×5 Punkte)**

Kreuze jeweils alle korrekten Aussagen an. Es gibt nur Punkte für vollständig korrekt angekreuzte Teilaufgaben.
(Hinweis: In jeder Teilaufgabe ist immer mindestens eine Aussage korrekt.)

a) Die Codierungsgröße einer Adjazenzmatrix eines Graphen mit n Knoten und m Kanten ist immer ...

- $\mathcal{O}(n^2)$. ☐
- $\mathcal{O}(m \log n)$. ☐
- $\mathcal{O}(m^2)$. ☐

b) Alle AVL-Bäume mit n Knoten und Höhe h ...

- besitzen mindestens $h-1$ Knoten. ☐
- benötigen $\mathcal{O}(1)$ RESTRUCTURE-Operationen nach einer Einfügeoperation. ☐
- benötigen $\Omega(n)$ RESTRUCTURE-Operationen nach einer Löschoperation. ☐

c) In welchen Fällen besitzt Quicksort eine Laufzeit von $\mathcal{O}(n^2)$?

- Im Best Case. ☐
- Im Average Case. ☐
- Im Worst Case. ☐

d) Die Breitensuche ...

- eignet sich zur Bestimmung von kürzesten Wegen in Graphen. ☐
- nutzt eine Warteschlange als Datenstruktur. ☐
- hat eine Laufzeit von $\mathcal{O}(n)$. ☐

e) Ein zusammenhängender Graph mit $n > 2$ Knoten besitzt immer einen Hamiltonkreis, wenn jeder Knoten den Grad ...

- 2 besitzt. ☐
- mindestens 4 besitzt. ☐
- $n-1$ besitzt. ☐

Viel Erfolg ☺