Technische Universität Braunschweig
Institut für Betriebssysteme und Rechnerverbund
Abteilung Algorithmik

Wintersemester 2018/2019

Prof. Dr. Sándor P. Fekete
Christian Rieck
Arne Schmidt

Klausur
Algorithmen und Datenstrukturen
22.02.2019

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	**17**	**12**	**17**	**12**	**10**	**11**	**11**	**10**	**100**
Erreicht									
Note	—	—	—	—	—	—	—	—	

Aufgabe 1: Euler und Hamilton **(8+4+5 Punkte)**

a) Wende den Algorithmus von Hierholzer auf den Graphen G aus Abbildung 1 an; starte dabei mit dem Knoten v_1. Kommen in einem Schritt des Algorithmus mehrere Knoten in Frage, wähle denjenigen mit dem kleinsten Index. Gib in jeder Iteration den gefundenen Kreis als Knotenliste an und markiere in den Listen, wenn Kreise verschmolzen werden.

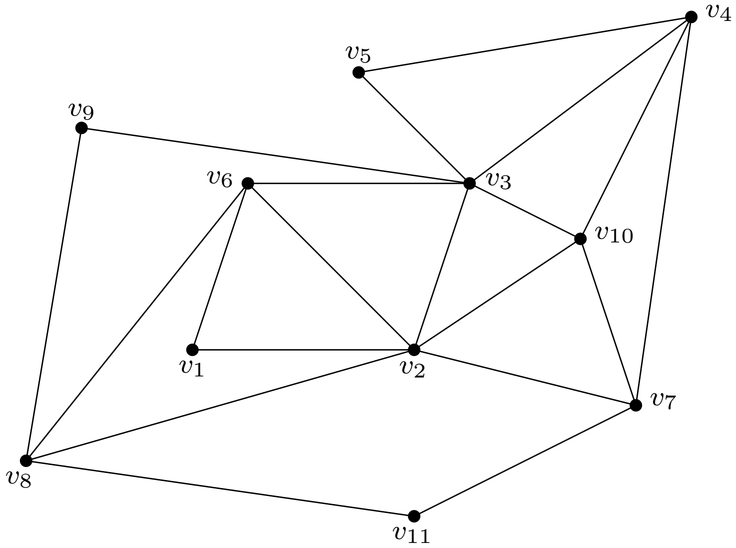


Abbildung 1: Der Graph G.

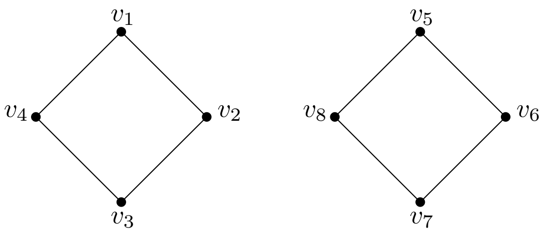


Abbildung 2: Der Graph H.

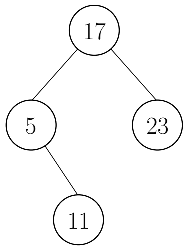
b) Gegeben sei der Graph H aus Abbildung 2. Füge so wenig Kanten wie möglich zu H hinzu, sodass der resultierende einfache Graph eulersch ist. Begründe, warum der resultierende Graph eulersch ist und warum es nicht weniger Kanten sein können! (Hinweis: Die Kanten können direkt in den Graphen aus Abbildung 2 eingezeichnet werden.)

c) Sei I ein Graph mit mindestens 3 Knoten. Zeige: Existiert in I ein Hamiltonkreis, dann zerfällt I in höchstens k Komponenten, wenn k Knoten entfernt werden.

Aufgabe 2: AVL-Bäume **(2+2+3+5 Punkte)**

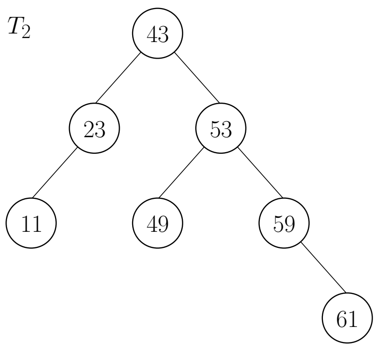
a) Führe INSERT(T_1, 7) auf dem AVL-Baum aus. Gib den Baum nach der Einfügeoperation sowie nach jeder Rotationsoperation an. (Hinweis: Die Einfügeoperation darf in dem angegebenen Baum durchgeführt werden.)

T_1

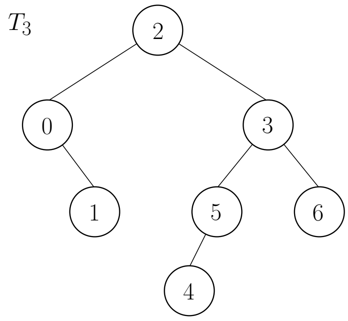


b) Führe DELETE(T_2, 23) auf dem AVL-Baum aus. Gib den Baum nach der Löschoperation sowie nach jeder Rotationsoperation an.

T_2



c) Argumentiere, warum T_3 kein AVL-Baum ist. Gib einen AVL-Baum minimaler Höhe mit den Zahlen in T_3 an.



d) Sei B ein voller binärer Suchbaum und n_I die Anzahl an inneren Knoten in B. Zeige oder widerlege: B besitzt genau $n_I + 1$ Blätter.

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

a) Kreuze an, in welchen Klassen die jeweilige Funktion liegt (ohne Begründung).

$f(n)$	$\mathcal{O}(n)$	$\Omega(1)$	$\Omega(n \log n)$	$\mathcal{O}(n \log n)$	$\mathcal{O}(n^2)$
$\log_2(n!)$					
$\frac{n^2}{\log n}$					
$\frac{\log n}{n^2}$					
$(2 + \cos(\pi n))^n$					

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

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

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

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

e) Betrachte die Funktion $x(n) = (1 - (-1)^n)^n$.
Zeige oder widerlege: Die Funktion $x(n)$ liegt in $\Theta(2^n)$.

Aufgabe 4: Rekursionen **(3+3+3+3 Punkte)**

Bestimme mithilfe des Mastertheorems das asymptotische Wachstum der Rekursionen aus den Aufgabenteilen a) - d), falls das Mastertheorem anwendbar ist oder begründe, warum dieses nicht funktioniert. Ist das Mastertheorem anwendbar, bestimme die Werte aller auftretenden Parameter.

a)

$$W(n) = 8 \cdot W\left(\frac{n}{4}\right) + 4n^2 + 16 \cdot W\left(\frac{n}{8}\right) + W\left(\frac{n}{2}\right) - 17n$$

b)

$$T(n) = 3 \cdot T\left(\frac{n}{25}\right) + 2\log n + 2 \cdot T\left(\frac{n}{100}\right) + 7\sqrt{n}$$

c)

$$U(n) = 23 \cdot U\left(\frac{n}{7}\right) + n^2 - 7 \cdot U\left(\frac{n}{2}\right) - 23n$$

d)

$$V(n) = -5n^2 + 7 + 81 \cdot V\left(\frac{n}{3}\right) + 7n^3$$

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

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

b) Welche Schritte werden benötigt, um das Rang-k Element in einer Menge von n paarweise verschiedenen Zahlen in $O(n)$ Zeit zu finden? Kreuze in jeder Teilaufgabe den richtigen Schritt an.

(i) Teile die n Zahlen in t viele
- ◯ 1er Gruppen auf.
- ◯ 3er Gruppen auf.
- ◯ 5er Gruppen auf.

(ii) In jeder Gruppe j $(1 \leq j \leq t)$ sucht man
- ◯ das Minimum m_j.
- ◯ den Median m_j.
- ◯ das Maximum m_j.

(iii) Sei M die Menge aller m_j mit $1 \leq j \leq t$. Suche in M
- ◯ das Rang-$\lceil \sqrt{t} \rceil$ Element x.
- ◯ das Rang-$\lceil \frac{t}{2} \rceil$ Element x.
- ◯ das Rang-$\lceil \log_2 t \rceil$ Element x.

(iv) Angenommen, es gibt nun $(i-1)$ Zahlen kleiner als x und es gilt $k > i$. Um das Rang-k Element zu finden, suchen wir in der Menge der Zahlen größer als x rekursiv nach dem
- ◯ Rang-$(k-i)$ Element.
- ◯ Rang-k Element.
- ◯ Rang-$(n-i)$ Element.

c) Zeige, dass das in der Vorlesung vorgestellte Verfahren zum Finden des Rang-k Elements eine Laufzeit von $O(n)$ besitzt.

Aufgabe 6: Sortieren **(6+3+2 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 ist für Nebenrechnungen gedacht.)

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

Abbildung 3: Quicksort im Array A.

b) Es ist bekannt, dass QUICKSORT aus der Vorlesung im Worst-Case eine Laufzeit von $\Theta(n^2)$ besitzt. Zeige oder widerlege: Es kann in PARTITION immer ein Pivotelement gewählt werden, sodass für QUICKSORT eine Laufzeit von $\Theta(n \log n)$ garantiert werden kann.

c) Welche Eigenschaft erfüllt ein stabiler Sortieralgorithmus? Ist Quicksort in der aus der Vorlesung bekannten Version stabil? Begründe deine Antwort!

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

Betrachte eine Menge von n paarweise verschiedenen ganzen Zahlen in einem Array A.

a) Zeige: Algorithmus 1 entscheidet in $O(n \log n)$ Zeit, ob zwei verschiedene Zahlen in A existieren, die sich zu einer gegebenen Zahl K addieren.
(Hinweis: Hier ist ein Korrektheitsbeweis und eine Laufzeitanalyse gefordert!)

```
function TwoSum(A, K)
    Mergesort(A, 1, n)
    i ← 1, j ← n
    while i < j do
        if A[i] + A[j] = K then
            return true
        else if A[i] + A[j] < K then
            i ← i + 1
        else
            j ← j − 1
    return false
```

Algorithmus 1: Ein Algorithmus für das Problem in Teilaufgabe 7a).

b) Beschreibe, wie der Algorithmus aus Aufgabenteil a) genutzt werden kann, um in $O(n^2)$ Zeit zu entscheiden, ob drei verschiedene Zahlen existieren, deren Summe 0 ergibt. (Hinweis: Ein Korrektheitsbeweis ist nicht notwendig.)

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

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

a) Für welche Graphenklassen ist der BFS-Baum immer eindeutig, wenn bei einem festen Knoten gestartet wird?

- Vollständige Graphen ☐
- Bäume ☐
- Kreise gerader Länge ☐

b) Sei $G = (V, E)$ ein Graph ohne isolierte Knoten. Welche Eigenschaften muss G besitzen, damit er eine Eulertour besitzt?

- Jeder Knoten muss geraden Grad besitzen. ☐
- Der Graph benötigt mindestens $|V|$ Kanten. ☐
- Der Graph muss zusammenhängend sein. ☐

c) AVL-Bäume der Höhe h...

- ... besitzen immer $2^h - 1$ Knoten. ☐
- ... besitzen maximal 2^{h-1} Blätter. ☐
- ... benötigen $O(1)$ Rotationsoperationen nach einer Löschoperation. ☐

d) Welche Laufzeitschranken sind korrekt?

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

e) In welche Datenstruktur kann in $O(1)$ Zeit ein Element eingefügt werden?

- Binärer Suchbaum ☐
- Warteschlange/Queue ☐
- Doppelt verkettete Liste ☐

Viel Erfolg ☺