Technische Universität Braunschweig
Institut für Betriebssysteme und Rechnerverbund
Abteilung Algorithmik

Sommersemester 2019

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

Klausur
Algorithmen und Datenstrukturen
10.08.2019

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 12 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	**19**	**11**	**18**	**12**	**8**	**11**	**11**	**10**	**100**
Erreicht									
Note	—	—	—	—	—	—	—	—	

Aufgabe 1: Euler und Hamilton **(6+4+4+5 Punkte)**

a) Benutze den Algorithmus von Fleury zum Bestimmen einer Eulertour/eines Eulerweges auf den Graphen G aus Abbildung 1 an. Gib dazu den abgelaufenen Weg als Knotenfolge an. Kommen in einem Schritt des Algorithmus mehrere Knoten infrage, wähle denjenigen mit dem kleinsten Index.

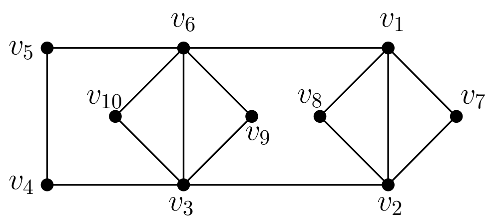


Abbildung 1: Der Graph G.

b) Besitzt der Graph aus Abbildung 1 einen Hamiltonkreis? Begründe deine Antwort!

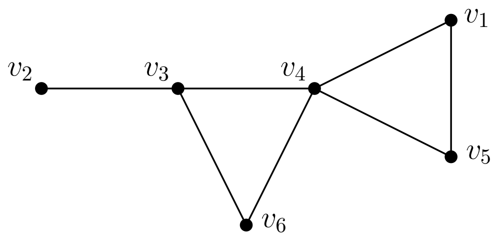


Abbildung 2: Der Graph H.

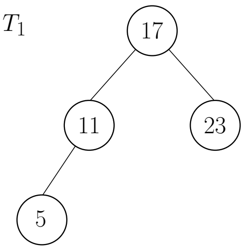
c) Betrachte den Graphen H aus Abbildung 2. Füge zwei Kanten in H ein, sodass der resultierende, einfache Graph sowohl eine Eulertour als auch einen Hamiltonkreis enthält. Begründe außerdem warum er beide Eigenschaften enthält und warum es nicht ausreicht, eine Kante einzufügen.

d) Beweise: Besitzt ein Graph I eine Brücke, so gibt es in I mindestens einen Knoten mit ungeradem Grad.
(Hinweis: Eine Kante e eines Graphen heißt Brücke, wenn der Graph ohne e unzusammenhängend ist.)

Aufgabe 2: Binäre Suchbäume und Heaps **(2+2+4+3 Punkte)**

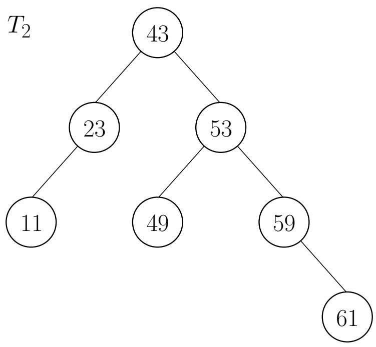
a) Führe INSERT(T_1, 7) auf dem AVL-Baum T_1 aus. Gib den Baum nach der Einfügeoperation sowie nach jeder Rotationsoperation an, sodass am Ende wieder ein AVL-Baum entsteht.

T_1



b) Führe DELETE(T_2, 43) auf dem AVL-Baum T_2 aus. Gib den Baum nach der Löschoperation sowie nach jeder Rotationsoperation an, sodass am Ende wieder ein AVL-Baum entsteht.

T_2



c) Zeige per Induktion, dass ein AVL-Baum der Höhe h mindestens $F_{h+2} - 1$ Knoten besitzt. Dabei bezeichnet F_n die n-te Fibonacci-Zahl.

d) Sei $M := [10, 7, 9, 6, 2, 4, 8, 5, 3, 0, 1]$ ein Max-Heap. Gib M in Baumstruktur an.

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

a) 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
$\Omega(n)$		$\Theta(\log^2(n))$
$O(n)$		$O(\sqrt{n})$
$O\left(\sum_{i=1}^{n} i\right)$		$O(n^2)$
$\Omega(n)$		$\Omega\left(\frac{n}{\log n}\right)$
$O\left(\frac{1}{n}\right)$		$\Theta\left(\frac{1}{n^2}\right)$
$O\left(\sum_{i=1}^{n} i^3\right)$		$O\left(\left(\sum_{i=1}^{n} i\right)^2\right)$

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

c) Zeige oder widerlege: Die Funktion $\log_2(n!)$ liegt in $\Omega(n \log_2 n)$.

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)

$$W(n) = 7 \cdot W\left(\frac{n}{4}\right) + 3n^2 + 16 \cdot W\left(\frac{n}{8}\right) + W\left(\frac{n}{2}\right) - 17 \log n$$

b)

$$T(n) = 3 \cdot T\left(\frac{n}{25}\right) + 2 \log n + 4 \cdot T\left(\frac{n}{100}\right) + 7\sqrt{n}$$

c)

$$U(n) = U\left(\frac{n}{4}\right) + n^2 + \sum_{i=1}^{5} U\left(\frac{n}{i^2}\right) - 23n$$

d)

$$V(n) = 5n + 7 + 64 \cdot V\left(\frac{n}{4}\right) + 7n^2$$

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

Wir betrachten eine Menge X von n paarweise verschiedenen Zahlen.

a) Welche Schritte werden benötigt, um das Rang-k Element in X in $O(n)$ Zeit zu finden? Kreuze in jeder Teilaufgabe den richtigen Schritt an.

(i) Teile die n Zahlen in t viele
- ◯ 5er Gruppen auf.
- ◯ 4er Gruppen auf.
- ◯ 3er Gruppen auf.

(ii) In jeder Gruppe j $(1 \leq j \leq t)$ sucht man
- ◯ das Rang-1 Element m_j.
- ◯ das Rang-2 Element m_j.
- ◯ das Rang-3 Element m_j.

(iii) Sei M die Menge aller m_j mit $1 \leq j \leq t$. Suche in M
- ◯ das Rang-$\lceil \frac{t}{2} \rceil$ Element x.
- ◯ das Rang-$\lceil \frac{t}{4} \rceil$ Element x.
- ◯ das Rang-$\lceil \frac{t}{8} \rceil$ 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 kleiner als x rekursiv nach dem
- ◯ Rang-$(k-i)$ Element.
- ◯ Rang-k Element.
- ◯ Rang-$(n-i)$ Element.

b) Beschreibe, wie die ℓ größten Zahlen in X in $O(n)$ Zeit gefunden werden können.

Aufgabe 6: Sortieren **(6+3+2 Punkte)**

a) Sortiere das Array A aus Abbildung 3 mit dem Algorithmus MergeSort. Gib (separat und in chronologischer Reihenfolge) 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 3 gegebene Tabelle.

$A =$	12	28	15	3	21	17	4	9
1. $A =$								
2. $A =$								
3. $A =$								
4. $A =$								
5. $A =$								
6. $A =$								
7. $A =$								

Abbildung 3: Mergesort im Array A.

b) Sei A ein Array mit paarweise verschiedenen Elementen. 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?

Aufgabe 7: Algorithmenentwurf **(3+8 Punkte)**

Das *Zentrum* eines Baumes ist ein Knotenmenge V_Z, sodass für jeden Knoten $v \in V_Z$ $\max_{w \in V} d(v, w)$ kleinstmöglich ist. Dabei bezeichnet $d(v, w)$ die Länge des eindeutigen Pfades von v zu w. Man kann zeigen, dass das Zentrum eines Baumes aus einem oder zwei adjazenten Knoten besteht.

a) Markiere alle Knoten, die das Zentrum des Baumes aus Abbildung 4 bilden.

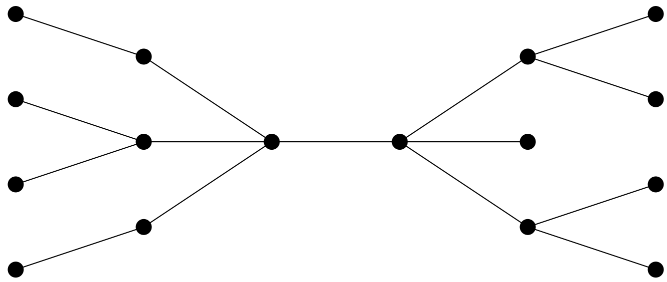


Abbildung 4: Ein Baum T.

b) Eine Möglichkeit, das Zentrum zu finden, ist das sukzessive Entfernen aller Blätter, bis nur noch ein oder zwei Knoten übrig sind. Entwirf einen Algorithmus (in Pseudocode), der das Zentrum eines gegebenen Baumes in $O(n)$ Zeit bestimmt.

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) Der BFS-Algorithmus. . .

- . . . benötigt $O(n^2)$ Zeit. ☐
- . . . benutzt eine Warteschlange als Datenstruktur. ☐
- . . . findet kürzeste Wege in ungewichteten Graphen. ☐

b) In einer doppelt verketteten Liste. . .

- . . . dauert eine Einfügeoperation $O(1)$ Zeit. ☐
- . . . sind die Elemente sortiert. ☐
- . . . dauert das Suchen nach einem Element $O(n)$ Zeit. ☐

c) Sei A ein unsortiertes Array mit n Elementen. Dann kann man in $O(n)$ Zeit. . .

- . . . den Median in A bestimmen. ☐
- . . . einen Max-Heap aus A konstruieren. ☐
- . . . einen AVL-Baum aus A konstruieren. ☐

d) Welche der folgenden Eigenschaften erfüllt ein Graph, der einen Hamiltonkreis beinhaltet, auf jeden Fall?

- Er besitzt mindestens $2n$ Kanten. ☐
- Er ist zusammenhängend. ☐
- Er besitzt genau einen Kreis. ☐

e) Welche Aussagen zu Sortierverfahren sind korrekt?

- Die Laufzeit von Mergesort ist $O(n \log n)$ ☐
- Mergesort ist ein stabiles Sortierverfahren. ☐
- Quicksort vergleicht $O(n \log n)$ Zahlen. ☐

Viel Erfolg ☺