Technische Universität Braunschweig
Institut für Betriebssysteme und Rechnerverbund
Abteilung Algorithmik

Wintersemester 2022/2023

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

Klausur
Algorithmen und Datenstrukturen
03.03.2023

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	**15**	**15**	**15**	**16**	**10**	**12**	**7**	**10**	**100**
Erreicht									
Note	—	—	—	—	—	—	—	—	

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

a) Betrachte den Graphen G aus Abbildung 1. Wende den Algorithmus von Fleury zum Bestimmen eines Eulerwegs auf G an und gib den Weg als Knotenliste an. Stehen zu einem Zeitpunkt mehrere Knoten zur Auswahl, wähle unter diesen den Knoten mit dem kleinsten Index.

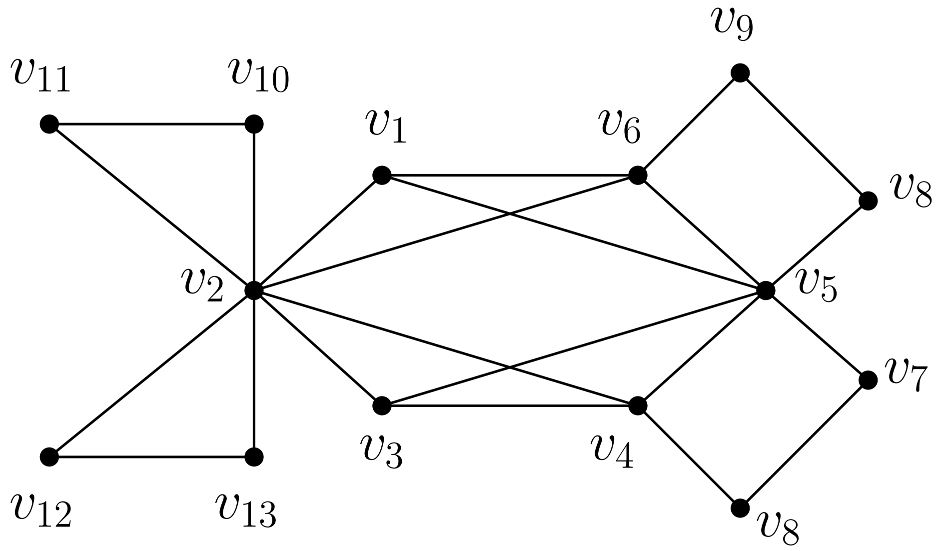


Abbildung 1: Der Graph G

b) Welche zwei Eigenschaften müssen in einem Graphen G vorliegen, damit in G eine Eulertour (kein Eulerweg) existiert?

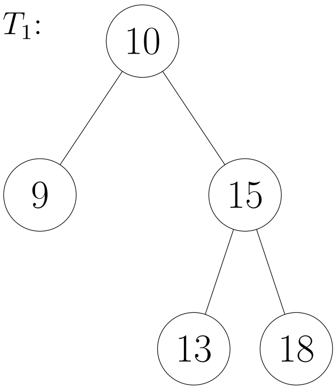
c) Zeige: Jeder zusammenhängende Graph kann so mit Kanten (keine Knoten) erweitert werden, sodass der erhaltene, nicht notwendigerweise einfache Graph eine Eulertour enthält.

d) Gilt c) weiterhin, wenn der erhaltene Graph einfach sein muss? Begründe deine Antwort.

Aufgabe 2: Dynamische Datenstrukturen **(2+2+5+3+3 Punkte)**

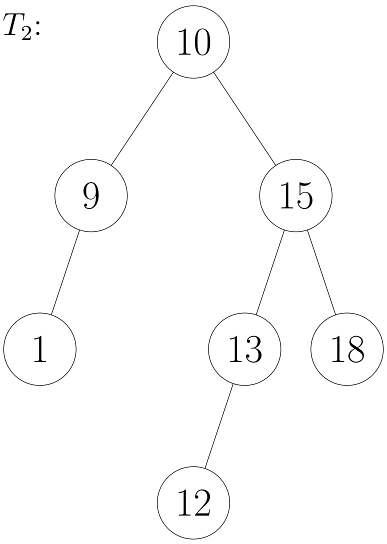
a) Führe Insert(T_1, 14) auf dem AVL-Baum aus. Gib den Baum nach der Einfügeoperation sowie nach jeder Restructure-Operation an.
(Hinweis: Die Einfügeoperation darf in dem angegebenen Baum durchgeführt werden.)

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) Zeige, dass das Erstellen eines binären Suchbaums aus einer unsortierten Menge von vergleichbaren Elementen $\Omega(n \log n)$ Zeit benötigt.
(Hinweis: Die Inorder-Ausgabe eines binären Suchbaums benötigt $O(n)$ Zeit.)

d) Wie ist ein Max-Heap definiert?

e) Sei $M = [12, 9, 11, 8, 5, 7, 10, 3, 4, 2, 1, 6]$ ein Max-Heap. Gib M in Baumstruktur an.

Aufgabe 3: Wachstum von Funktionen **(4+5+6 Punkte)**

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

b) Zeige: $\log(n!) \in \Omega(n \log n)$.

c) Ordne folgende Klassen nach Inklusion. Markiere außerdem identische Klassen (zB durch ein Gleichheitssymbol).
(Hinweis: $A \subset B$, wenn jedes Element aus A auch in B vorkommt.)

$$O(15), \quad O(n \log n), \quad O\left(\frac{n}{5}\right), \quad O(\log n), \quad O(2^n), \quad O(n^2), \quad O(n - \log n), \quad O(3^n)$$

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

a) Wie lautet das Mastertheorem aus der Vorlesung?

b) Entscheide für die folgenden Rekursionen, ob das Mastertheorem anwendbar ist. Falls es angewendet werden kann, bestimme die Laufzeit der Rekursion und gib die im Mastertheorem auftretenden Parameter an.

Falls es nicht angewendet werden kann, gib eine Begründung dafür an.

(i)

$$T(n) = T\left(\frac{4n}{5}\right) + 9 \cdot T\left(\frac{n}{5}\right) + 7n^2 - n$$

(ii)

$$U(n) = 2 \cdot U\left(\frac{n}{2}\right) - U\left(\frac{n}{4}\right) + 4n$$

(iii)

$$V(n) = 2n^2 + 2 \cdot V\left(\frac{3n}{5}\right) + 4 \cdot V\left(\frac{n}{4}\right)$$

(iv)

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

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 $m \in X$?

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 gleichmäßig in
- ◯ $t = \lceil n/2 \rceil$ viele Gruppen auf.
- ◯ $t = \lceil n/3 \rceil$ viele Gruppen auf.
- ◯ $t = \lceil n/5 \rceil$ viele Gruppen auf.

(ii) Suche in jeder Gruppe j $(1 \leq j \leq t)$
- ◯ 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-$\lceil t/2 \rceil$ Element x.
- ◯ das Rang-$\lceil \sqrt{t} \rceil$ 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) Sei $\ell \in \{1, \ldots, n\}$ und X eine Menge von n paarweise verschiedenen Zahlen.

Zeige: Die ℓ kleinsten Zahlen aus X können in Zeit $O(n + \ell \log \ell)$ in sortierter Reihenfolge ausgegeben werden.

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

a) Sortiere das Array A aus Abbildung 2 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 2 gegebene Tabelle.

A	7	13	5	9	10	12	1	3
1. A								
2. A								
3. A								
4. A								
5. A								
6. A								
7. A								

Abbildung 2: MERGESORT auf dem Array A

b) Wie ist ein stabiler Sortieralgorithmus definiert? Gib außerdem einen stabilen und einen nicht stabilen Sortieralgorithmus an (ohne Begründung).

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

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

Sei B ein binärer Suchbaum mit Wurzel r. Wir wollen die Länge des längsten Pfades in B bestimmen, welcher in r startet. Die Länge eines Pfades P ist dabei die Anzahl an Knoten auf P.

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

(Hinweis: Es darf angenommen, dass der Algorithmus mit einem Aufruf von $\textsc{maxPath}(r)$ gestartet wird.)

Aufgabe 8: Kurzfragen **(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) Breitensuche...

- ...nutzt einen Stapel als Datenstruktur ☐
- ...findet kürzeste Wege. ☐
- ...besitzt Laufzeit $O(m+n)$ ☐

b) Quicksort...

- ...ist ein vergleichsbasierter Algorithmus. ☐
- ...nutzt das Prinzip „Teile und Herrsche" (Divide and Conquer). ☐
- ...besitzt Worst-Case-Laufzeit $O(n^2)$. ☐

c) Welche Aussagen zu AVL-Bäumen der Höhe h sind korrekt?

- $h \in O(\log n)$. ☐
- Eine Einfügeoperation erfordert höchstens eine Restructureoperation. ☐
- Das Suchen nach einem Element benötigt $O(h)$ Zeit. ☐

d) In welcher der folgenden Datenstrukturen können Elemente in $O(1)$ Zeit eingefügt werden?

- Einfach verkettete Listen. ☐
- Warteschlange ☐
- Binärer Suchbaum ☐

e) Wird eine Subroutine, welche die Laufzeit $O(n)$ besitzt, $O(\log n)$ mal wiederholt, so ist die asymptotische Gesamtlaufzeit...

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

Viel Erfolg ☺