Technische Universität Braunschweig
Institut für Betriebssysteme und Rechnerverbund
Abteilung Algorithmik

Wintersemester 2019/2020

Prof. Dr. Sándor P. Fekete
Arne Schmidt

Klausur
Algorithmen und Datenstrukturen
28.02.2020

Name: ....

Vorname: ....

Matr.-Nr.: ....

Studiengang: ....

☐ Bachelor ☐ Master ☐ Andere

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

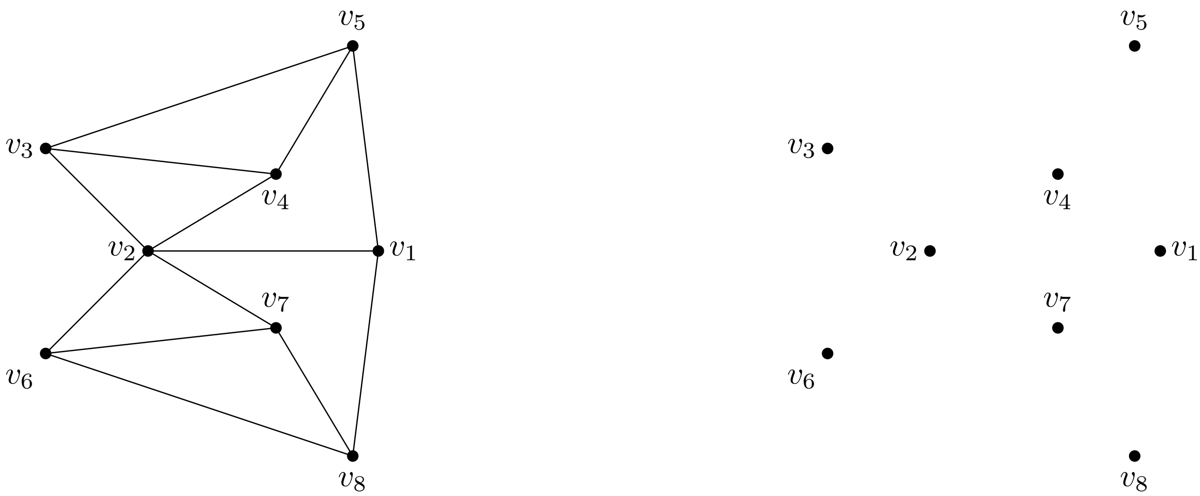
Hinweise:

- 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**	Σ
Punkte	**15**	**20**	**15**	**15**	**15**	**10**	**10**	**100**
Erreicht								
Note	—	—	—	—	—	—	—	

Aufgabe 1: Graphen (4+4+3+4 Punkte)

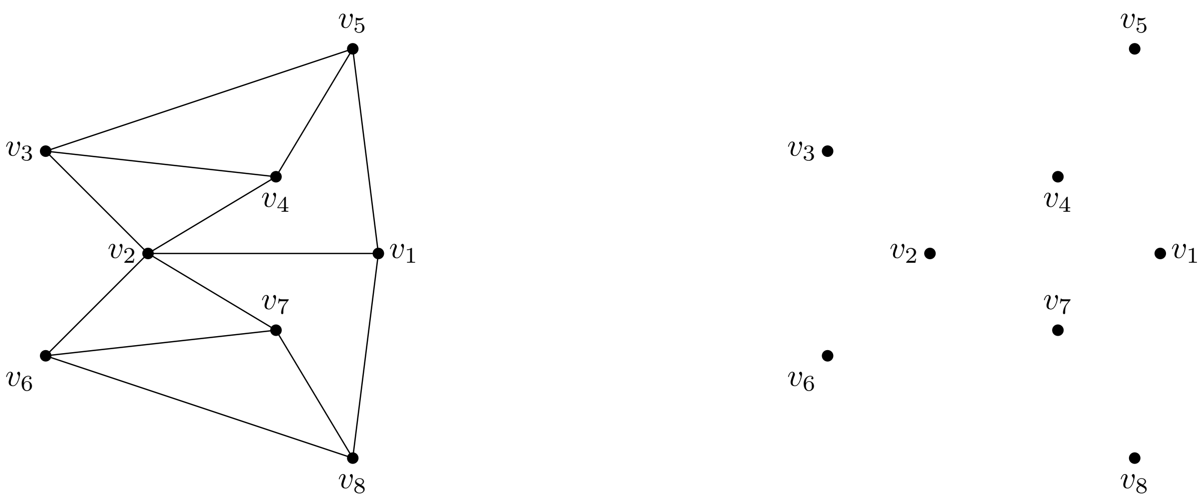
a) Betrachte den Graphen G aus Abbildung 1a. Zeichne die Kanten des **Breitensuchbaums** mit Startknoten v_1 in Abbildung 1b ein. Kommen in einem Schritt des Algorithmus mehrere Knoten in Frage, wähle denjenigen mit dem kleinsten Index. (Hinweis: Uns reicht die Angabe des Baums.)



(a) Der Graph G **(b)** Breitensuchbaum von G

Abbildung 1

b) Betrachte den Graphen G aus Abbildung 2a. Zeichne die Kanten des **Tiefensuchbaums** mit Startknoten v_1 in Abbildung 2b ein. Kommen in einem Schritt des Algorithmus mehrere Knoten in Frage, wähle denjenigen mit dem kleinsten Index. (Hinweis: Uns reicht die Angabe des Baums.)



(a) Der Graph G **(b)** Tiefensuchbaum von G

Abbildung 2

c) Welche Datenstrukturen benutzen Breiten- und Tiefensuche und welche Laufzeit besitzen die beiden Algorithmen?

Datenstruktur Breitensuche: ____________________

Datenstruktur Tiefensuche: ____________________

Laufzeit von Breiten- und Tiefensuche: ____________________

d) Betrachte den Graphen G aus Abbildung 3. Füge eine minimale Anzahl an Kanten ein, sodass im resultierenden (einfachen) Graphen ein (nicht notwendig geschlossener) **Eulerweg** existiert. Begründe außerdem, warum weniger Kanten nicht ausreichen.

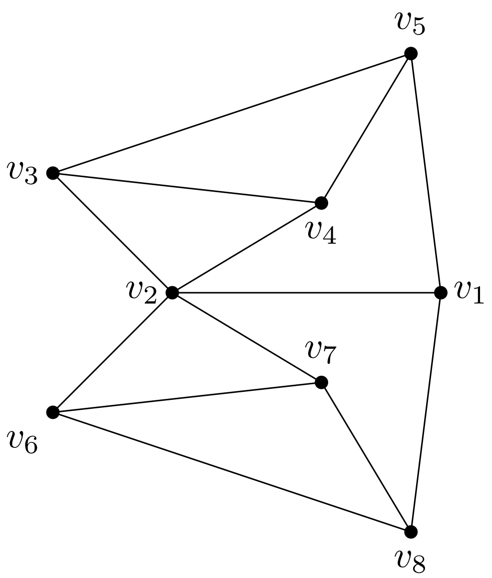


Abbildung 3: Der Graph G

Aufgabe 2: Dynamische Datenstrukturen (1+2+3+2+3+4+5 Punkte)

a) Betrachte die auf einem Array implementierte Warteschlange Q aus Abbildung 4. Liste alle Elemente auf, die in Q enthalten sind.

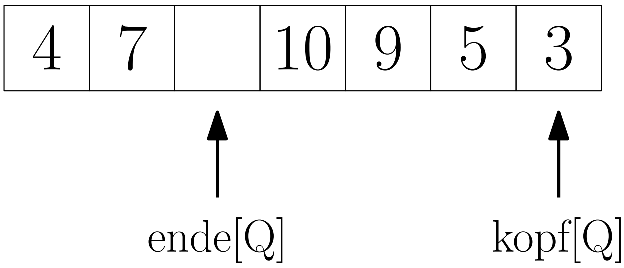


Abbildung 4: Die auf einem Array implementierte Warteschlange Q

In Q enthaltene Elemente: ____________________

b) Betrachte weiterhin die Warteschlange Q aus Abbildung 4. Führe die Operation $\textsc{Enqueue}(Q, 17)$ aus. Gib das Array (inklusive *kopf*- und *ende*-Zeiger) nach Durchführung der Operation an, indem du Abbildung 5 ausfüllst.

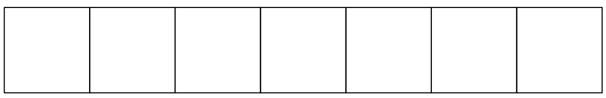
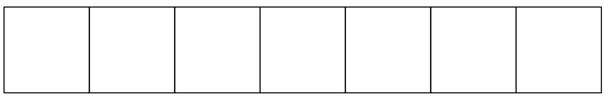


Abbildung 5: Warteschlange Q nach der Operation $\textsc{Enqueue}(Q, 7)$.

c) Betrachte die Warteschlange Q aus Abbildung 4. Führe die Operation $\textsc{Dequeue}(Q)$ aus. Gib das Array (inklusive *kopf*- und *ende*-Zeiger) nach Durchführung der Operation und das zurückgegebene Element an, indem du Abbildung 6 ausfüllst.

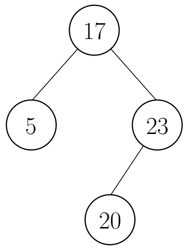


Rückgabe:

Abbildung 6: Warteschlange Q nach der Operation $\textsc{Dequeue}(Q)$.

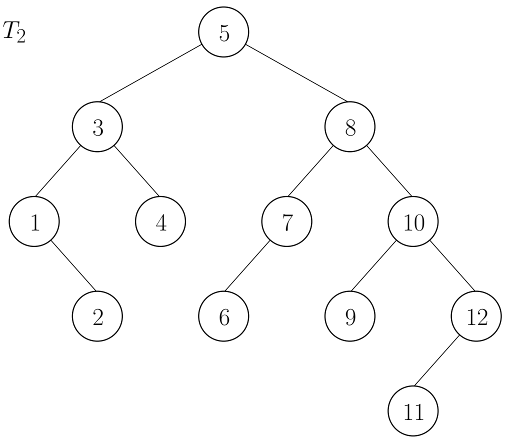
d) Führe INSERT(T_1, 21) auf dem AVL-Baum aus. Gib den Baum nach der Einfügeoperation sowie nach jeder Restructure-Operation an.
(Hinweis: Die Einfügeoperation darf in dem angegebenen Baum durchgeführt werden.)

T_1



e) Führe DELETE(T_2, 4) auf dem AVL-Baum aus. Gib den Baum nach der Löschoperation sowie nach jeder Restructure-Operation an.

T_2



f) Zeige mit vollständiger Induktion, dass ein AVL-Baum der Höhe h maximal $2^h - 1$ Knoten besitzt.

g) 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.)

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

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

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

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

Aufgabe 4: Rekursionen **(4+2+3+3+3 Punkte)**

a) Wie lautet das Mastertheorem aus der Vorlesung?

b) Auf welche der folgenden Rekursionsgleichungen kann das Mastertheorem angewendet werden? Eine Begründung ist nicht erforderlich.

(i) $T(n) = 3 \cdot T\left(\frac{n}{3}\right) + 2n - T\left(\frac{n}{9}\right)$

☐ wahr
☐ falsch

(ii) $T(n) = n \cdot T(\frac{n}{2}) + n$

☐ wahr
☐ falsch

c) Bestimme mit Hilfe des Mastertheorems das asymptotische Wachstum der Rekursion

$$T(n) = 3 \cdot T\left(\frac{n}{3}\right) + 2\sqrt{n} + 54 \cdot T\left(\frac{n}{9}\right) + 7n^2$$

Bestimme zusätzlich die Werte aller auftretenden Parameter.

d) Bestimme mit Hilfe des Mastertheorems das asymptotische Wachstum der Rekursion

$$U(n) = 7n + 64 \cdot U\left(\frac{n}{4}\right) - 21\log n - 5 + n^2$$

Bestimme zusätzlich die Werte aller auftretenden Parameter.

e) Bestimme mit Hilfe des Mastertheorems das asymptotische Wachstum der Rekursion

$$T(n) = T\left(\frac{n}{9}\right) + 5 \cdot T\left(\frac{n}{81}\right) + 7\sqrt{n}$$

Bestimme zusätzlich die Werte aller auftretenden Parameter.

Aufgabe 5: Mediane **(1+4+5+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) Zeige: Der Algorithmus zum Finden eines Rang-k Elements einer Menge X benötigt $O(n)$ Zeit.

d) Sei X eine Menge von paarweise verschiedenen Elementen.
Zeige: Die ℓ kleinsten Elemente in X können in $O(n)$ Zeit gefunden werden.

Aufgabe 6: Sortieren **(7+2+1 Punkte)**

a) Wir betrachten Radixsort zum Sortieren von acht in Abbildung 7 angegebenen Zahlen mit jeweils vier Ziffern. Sortiere das Array A aus Abbildung 7 mit Radixsort. Gib A nach jeder Iteration an – nutze dazu Abbildung 7.

1234				
1642				
0146				
2471				
8182				
1024				
8568				
1991				

Abbildung 7: Radixsort auf Array A (linke Spalte).

b) Welche Eigenschaft muss der in Radixsort verwendete Sortieralgorithmus besitzen? Gib außerdem die Definition dieser Eigenschaft an.

c) Welche Laufzeit besitzt Radixsort für n Zahlen mit je d Ziffern?
(Hinweis: Eine Angabe ohne Begründung ist ausreichend.)

Aufgabe 7: 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) In welcher Datenstruktur kann das größte Element in $O(1)$ Zeit gefunden werden?

- AVL-Baum ☐
- Max-Heap ☐
- Stapel ☐

b) Jeder Pfad eines Graphen G...

- ...besucht keinen Knoten doppelt. ☐
- ...besucht jede Kante höchstens einmal. ☐
- ...ist ein Hamiltonpfad. ☐

c) Es gibt AVL-Bäume der Höhe h, die...

- ...3^h Knoten besitzen. ☐
- ...$\Omega(h)$ Restructureoperationen nach einer Einfügeoperation benötigen. ☐
- ...$\Omega(h)$ Restructureoperationen nach einer Löschoperation benötigen. ☐

d) 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: $O(n+m)$ ☐

e) In welchen Fällen besitzt Quicksort die Laufzeit $\Theta(n^2)$?

- Best Case ☐
- Average Case ☐
- Worst Case ☐

Viel Erfolg ☺