Technische Universität Braunschweig
Institut für Betriebssysteme und Rechnerverbund
Abteilung Algorithmik

Wintersemester 2014/2015

Prof. Dr. Sándor P. Fekete
Dr. Christian Scheffer

Klausur
Algorithmen und Datenstrukturen
24. 02. 2015

Name: ....

Vorname: ....

Matr.-Nr.: ....

Studiengang: ....

☐ Bachelor ☐ Master ☐ Andere

Mit der Veröffentlichung meines Klausurergebnisses unter meiner Matrikelnummer bin ich einverstanden.

....
Unterschrift

Hinweise:

- Bitte das Deckblatt vollständig ausfüllen.
- Die Klausur besteht aus 15 Blättern, bitte auf Vollständigkeit überprüfen.
- Erlaubte Hilfsmittel: Keine.
- Eigenes Papier ist nicht erlaubt.
- Die Rückseiten der Blätter dürfen beschrieben werden.
- 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.
- Die Bearbeitungszeit für die Klausur beträgt 120 Minuten.

Aufgabe	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Σ**
Punkte	**16**	**10**	**13**	**18**	**10**	**10**	**9**	**14**	**100**
Erreicht									
Note	—	—	—	—	—	—	—	—	

Aufgabe 1: Graphen **(10+3+3 Punkte)**

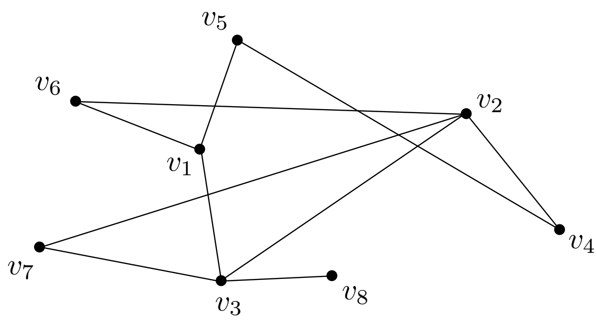


Abbildung 1: Der Graph G

a) Wende Breitensuche auf den Graphen G aus Abbildung 1 an; starte dabei mit dem Knoten v_1. Falls zu einem Zeitpunkt mehrere Knoten für den nächsten Schritt in Frage kommen, wähle denjenigen mit dem kleinsten Index. Gib die Menge R des Algorithmus Graph-Scan jedesmal an, wenn sie sich ändert, und zeichne den gefundenen Baum T.

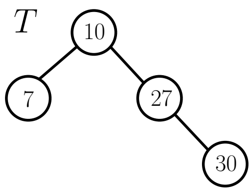
b) Zeichne einen Graphen mit 5 Knoten, der einen Eulerweg, aber keine Eulertour beinhaltet.

c) Zeige oder widerlege: Ein Graph, der nicht zusammenhängend ist besitzt, keinen Hamiltonpfad.

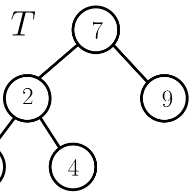
Aufgabe 2: AVL-Bäume **(2+2+2+2+2 Punkte)**

Betrachte in den Aufgabenteilen a) bis e) den Baum, der in der jeweiligen Abbildung dargestellt wird. Führe die, INSERT-Operation, die in dem jeweiligen Aufgabenteil genannt ist, und die damit verbundenen Restrukturierungsmaßnahmen (damit sind die Algorithmen aus der Vorlesung gemeint, die die AVL-Eigenschaft erhalten) auf dem entsprechenden Baum aus. Zeichne dabei das Resultat nach jeder einzelnen ausgeführten Operation INSERT und RESTRUCTURE in einen separaten Baum:

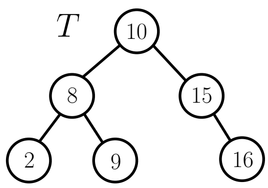
a) INSERT$(T, 52)$



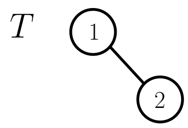
b) INSERT$(T, 5)$



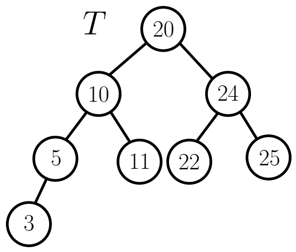
c) INSERT$(T, 29)$



d) $\textsc{Insert}(T, 5)$



e) $\textsc{Insert}(T, 1)$



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

Seien $f, g, h : \mathbb{N} \to \mathbb{R}$ drei Funktionen.

a) Zeige oder widerlege: $(g)^2 \in \mathrm{O}(f) \Rightarrow g \in \Omega(f)$

b) Zeige oder widerlege: $g \in \Omega(f), f \in \Theta(h) \Rightarrow h \in \Theta(g)$

c) Zeige: $7n^4 + 2n^3 - 14 \in \Theta(n^4)$. Gib dazu explizit geeignete Konstanten c_1, c_2 und n_0 aus der Definition an und zeige, dass sie die Definition erfüllen.

d) Gesucht ist ein Algorithmus, der n Elemente bearbeiten soll. Dazu gibt es zwei Möglichkeiten:

(i) Die Elemente werden der Reihe nach bearbeitet, das dauert $\mathrm{O}(n^2)$ pro Element.

(ii) Die Elemente werden erst sortiert. Dadurch reduziert sich die Bearbeitungszeit pro Element auf $\mathrm{O}(n \log n)$.

Welche Laufzeiten ergeben sich für beide Varianten, wenn für (ii) ein möglichst schnelles Sortierverfahren verwendet wird? Begründe Deine Antwort. Welche Vorgehensweise ist asymptotisch schneller?

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

a) Wie lautet das Mastertheorem aus der Vorlesung?

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

$$T(n) = 27 \cdot T\left(\frac{n}{4}\right) + 3n + 37 \cdot T\left(\frac{n}{4}\right).$$

Bestimme die Werte aller im Mastertheorem auftretender Parameter.

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

$$U(n) = n^3 + 10 \cdot U\left(\frac{n}{2}\right) - 19 + 3 \cdot U\left(\frac{n}{\sqrt[4]{8}}\right) + 29n^4.$$

Bestimme die Werte aller im Mastertheorem auftretender Parameter.

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

$$V(n) = 3 \cdot V\left(\frac{n}{3}\right) + 2 \cdot V\left(\frac{n}{4}\right) + 7n^2.$$

Bestimme die Werte aller im Mastertheorem auftretender Parameter.

e) Was sind obere und untere Schranken für die Worst-Case-Laufzeit von Quicksort? Wie muss die Eingabe beschaffen sein, damit der Worst-Case erreicht wird (mit Begründung)?

Aufgabe 5: Hashing **(10 Punkte)**

Wir betrachten ein anfangs leeres Array A der Größe 7, es gibt also die Speicherzellen $\mathrm{A}[0], \mathrm{A}[1], \ldots, \mathrm{A}[6]$. In diesem führen wir offenes Hashing mit der folgenden Hashfunktion durch:

$$t(i, x) = (x \cdot i + x) \mod 7$$

Dabei ist x ein einzusetzender Schlüssel und i die Nummer des Versuches, x in eine unbesetzte Speicherzelle des Arrays zu schreiben, beginnend bei $i = 0$. Berechne zu jedem der folgenden Schlüssel die Position, die er in A bekommt:

$$2, \quad 9, \quad 4, \quad 11$$

Dabei sollen die Schlüssel in der gegebenen Reihenfolge eingefügt werden, und der Rechenweg soll klar erkennbar sein. Trage die Elemente in das Array in Abbildung 2 ein.

$A[0]$	$A[1]$	$A[2]$	$A[3]$	$A[4]$	$A[5]$	$A[6]$

Abbildung 2: Die Hashtabelle.

Aufgabe 6: Sortieren **(10 Punkte)**

$A[1]$	$A[2]$	$A[3]$	$A[4]$	$A[5]$	$A[6]$	$A[7]$	$A[8]$
7	9	31	45	2	18	43	82

Abbildung 3: Mergesort im Array A.

Wende die Funktion MERGE(A, 1, 4, 8) aus MERGESORT auf das gefüllte Array oben in Abbildung 3 an. Gib dabei die temporären Variablen n_1 und n_2, sowie die Felder L und R an (jeweils nur das Endresultat). Führe Schritt für Schritt das Mergen von L und R in A durch, in dem Du die unten stehende Tabelle ausführst (pro Schritt eine Zeile). Ein Schritt entspricht dem Überschreiben von $A[k]$.

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

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

Gegeben sei ein binärer Suchbaum B, der nur positive Elemente beinhaltet und n Knoten hat. Gesucht wird der größte Schlüssel, der durch 7 ganzzahlig teilbar ist. Falls keiner durch 7 teilbar ist, soll 0 zurückgegeben werden.

a) Gib dafür einen rekursiven Algorithmus über linken und rechten Teilbaum an, der Laufzeit $\mathrm{O}(n)$ hat.

b) Begründe, warum dein Algorithmus in $\mathrm{O}(n)$ liegt.

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

a) Ein einfacher Graph ist zusammenhängend. ☐ wahr ☐ falsch

b) Breitensuche und Tiefensuche liefern immer unterschiedliche Ergebnisse. ☐ wahr ☐ falsch

c) Ein Baum mit mindestens einem Knoten hat mindestens ein Blatt. ☐ wahr ☐ falsch

d) Das Master-Theorem lässt sich auf jede Rekursionsformel anwenden. ☐ wahr ☐ falsch

e) Ein Eulerweg besucht alle Knoten eines Graphen. ☐ wahr ☐ falsch

f) Mergesort ist ein Algorithmus, der Divide-&-Conquer anwendet. ☐ wahr ☐ falsch

g) Breitensuche wird mit dem Graphen-Scan-Algorithmus mittels eines Stapels realisiert. ☐ wahr ☐ falsch

Viel Erfolg ☺