Technische Universität Braunschweig
Institut für Betriebssysteme und Rechnerverbund
Abteilung Algorithmik

Wintersemester 2013/2014

Prof. Dr. Sándor P. Fekete
Stephan Friedrichs

Klausur
Algorithmen und Datenstrukturen
25. 02. 2014

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 14 Blättern, bitte auf Vollständigkeit überprüfen.
- Erlaubte Hilfsmittel: Keine.
- Eigenes Papier ist nicht erlaubt.
- Die Rückseiten der Blätter dürfen beschrieben werden.
- Die Heftung der Blätter darf nicht entfernt werden.
- Die Klausur ist mit 50 % der Punkte bestanden.
- Antworten die *nicht* gewertet werden sollen bitte deutlich durchstreichen.
- 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	**15**	**10**	**13**	**18**	**10**	**11**	**9**	**14**	**100**
Erreicht									
Note	—	—	—	—	—	—	—	—	

Aufgabe 1: Graphen **(9+3+3 Punkte)**

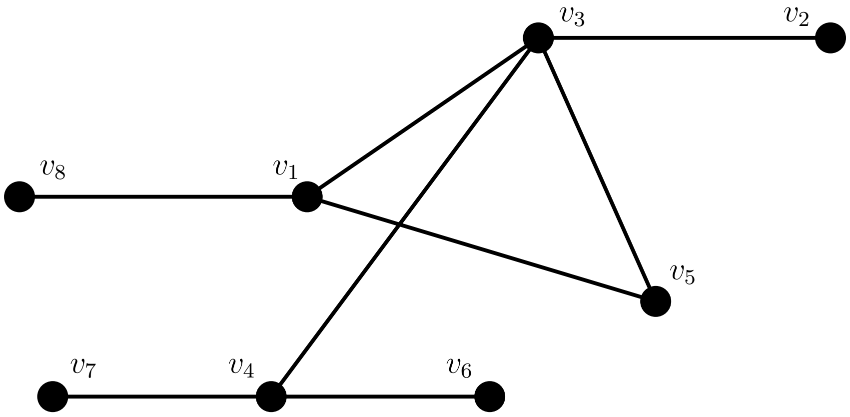

Abbildung 1: Der Graph G.

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 mindestens 8 Knoten, bei dem alle Knoten geraden Grad haben, der keinen Eulerweg besitzt.

c) Zeige oder widerlege: Eine Kantenfolge, die kein Weg ist, ist kein Pfad.

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

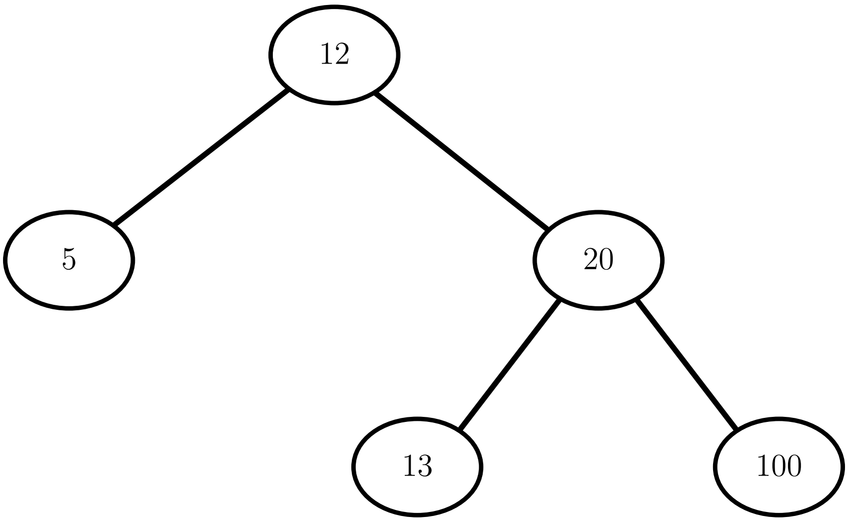

Abbildung 2: Der AVL-Baum T.

Gegeben ist der AVL-Baum T aus Abbildung 2. Führe nacheinander folgende Operationen aus (damit sind die Algorithmen aus der Vorlesung gemeint, die die AVL-Eigenschaft erhalten); zeichne dabei alle Schritte jeder Operation in einen separaten Baum:

a) $\textsc{Insert}(T, 58)$

b) $\textsc{Insert}(T, 4)$

c) $\textsc{Insert}(T, 3)$

d) $\textsc{Insert}(T, 2)$

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: $f \in \mathrm{O}(g), g \in \Omega(h) \Rightarrow h \in \mathrm{O}(f)$

b) Zeige oder widerlege: $g \in \mathrm{O}(f), f \in \Theta(h) \Rightarrow g \in \Theta(h)$

c) Zeige: $10n^3 - 5n^2 \log_4 n - 14 \in \Theta(n^3)$. 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 $\mathrm{O}(n)$ pro Element.

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

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

Aufgabe 4: Rekursionen **(4+4+3+3+1+3 Punkte)**

a) Wie lautet das Mastertheorem aus der Vorlesung?

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

$$T(n) = 8 \cdot T\left(\frac{n}{4}\right) + 3n + 32 \cdot T\left(\frac{n}{8}\right).$$

Bestimme die Werte aller im Mastertheorem auftretender Parameter.

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

$$U(n) = n + 14 \cdot U\left(\frac{n}{4}\right) - 2 + U\left(\frac{n}{\sqrt{8}}\right) + 4n^2.$$

Bestimme die Werte aller im Mastertheorem auftretender Parameter.

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

$$V(n) = 3 \cdot V\left(\frac{n}{8}\right) + 5 \cdot V\left(\frac{n}{8}\right) - n^2 + \frac{1}{4}n^3.$$

Bestimme die Werte aller im Mastertheorem auftretender Parameter.

e) Was ist die Worst-Case Laufzeit von QUICKSORT (ohne Begründung)?

f) Wir untersuchen eine Variante von QUICKSORT, die als Pivotelement den Median des zu sortierenden Teilbereichs verwendet. Der Median von n Elementen lässt sich in $\Theta(n)$ bestimmen. Benutze das Mastertheorem, um die Laufzeit dieser QUICKSORT-Variante zu bestimmen. (Hinweis: Beim Median handelt es sich um dasjenige Element, das das zu sortierende Feld in zwei gleich große Teilfelder zerlegt.)

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:

$$t(i, x) = \big(x \cdot i^2 + i\big) \mod 7$$

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:

$$3, \quad 12, \quad 4, \quad 14$$

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 3 ein.

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

Abbildung 3: Die Hashtabelle A.

Aufgabe 6: Sortieren **(11 Punkte)**

A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	A[10]	A[11]	A[12]	A[13]	A[14]	A[15]	A[16]
4	8	9	10	2	4	5	12	3	8	8	12	1	3	13	14

A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	A[10]	A[11]	A[12]	A[13]	A[14]	A[15]	A[16]

Abbildung 4: Mergesort im Array A.

Wende die Funktion $\text{Merge}(A, 9, 12, 16)$ aus Mergesort auf das gefüllte Array oben in Abbildung 4 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, mache dabei bei jedem Schritt kenntlich, woher das aktuelle Element stammt. Trage das Ergebnis in das leere Feld unten in Abbildung 4 ein.

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

Gegeben sei ein binärer Suchbaum B.

a) Gib einen Algorithmus an, der in $\mathrm{O}(n)$ alle Elemente von B in absteigend sortierter Reihenfolge ausgibt. Die Ausgabe eines Elements erfolgt mittels $\textsc{Print}(x)$.

b) Begründe, warum dein Algorithmus in $\mathrm{O}(n)$ liegt. Die Laufzeit von $\textsc{Print}$ liegt in $\mathrm{O}(1)$.

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

a) Die Tiefensuche beruht auf dem FIFO-Prinzip. ☐ wahr ☐ falsch

b) Breitensuche kann man auch parallel ausführen. ☐ wahr ☐ falsch

c) Gegeben zwei Funktionen f und g mit $f \in \mathrm{O}(g)$. Dann gilt für alle natürlichen Zahlen n, dass $f(n) \leq g(n)$. ☐ wahr ☐ falsch

d) Die Folge der Fibonacci-Zahlen wächst exponentiell, aber asymptotisch langsamer als 2^n. ☐ wahr ☐ falsch

e) Das Mastertheorem erlaubt das Lösen nichtlinearer Rekursionen. ☐ wahr ☐ falsch

f) Man kann schneller als in $\Omega(n \log n)$ sortieren, wenn man über geeignete Rechneroperationen verfügt. ☐ wahr ☐ falsch

g) Die Größe einer Inzidenzmatrix eines Graphen mit n Knoten ist $\mathrm{O}(n^2)$. ☐ wahr ☐ falsch

Viel Erfolg ☺