Technische Universität Braunschweig
IBR - Abteilung Algorithmik
Prof. Dr. Sándor Fekete
Dr. Christiane Schmidt
Stephan Friedrichs

Sommersemester 2013

Klausur
Algorithmen und Datenstrukturen
22.08.2013

Name: ....

Vorname: ....

Matr.-Nr.: ....

Studiengang: ....

Bachelor ☐ Master ☐ Diplom ☐ Andere ☐

Mit der Veröffentlichung des Klausurergebnisses nur mit der Matrikelnummer über die Mailingliste und auf der Homepage bin ich einverstanden.

....
Unterschrift

Hinweise:
· Bitte das Deckblatt ausfüllen. Die Heftung der Blätter darf nicht entfernt werden. Eigenes Papier ist nicht erlaubt. Die Rückseiten dieser Blätter dürfen beschrieben werden.
· Die Klausur besteht aus 15 Blättern.
· Hilfsmittel: keine.
· Die Klausur ist mit 50 von 100 Punkten bestanden.
· Alle Graphen in dieser Klausur sind einfache Graphen, d. h. sie haben keine Multikanten und keine parallelen Kanten; das gilt auch für die von Dir zu konstruierenden Graphen.
· Mit *Bleistift* oder *in rot* geschriebene Klausurteile können nicht gewertet werden.
· Die Bearbeitungszeit für die Klausur ist 120 Minuten.
· **Bearbeitete Aufgaben bitte unten ankreuzen.**

Punktzahlen für die Korrektur freilassen!

Aufgabe	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	Σ
Bearbeitet ($\times$)										
Punkte	**14**	**11**	**19**	**17**	**11**	**10**	**4**	**4**	**10**	**100**
Erzielte Punkte										

1.Aufgabe: Graphen **8+2+4 Punkte**

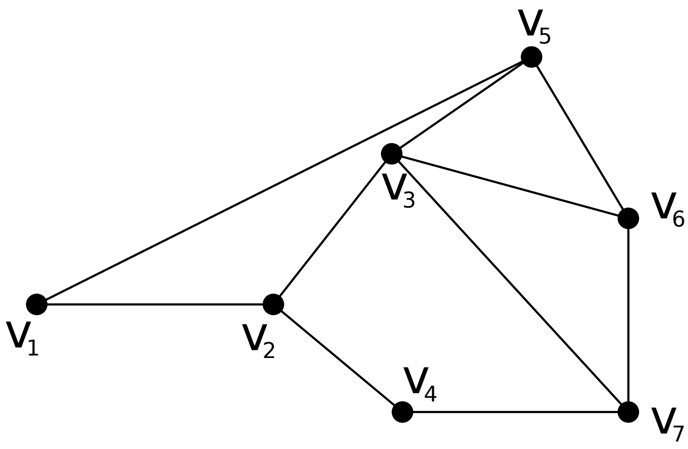

Abbildung 1: Der Graph H.

a) Wende Tiefensuche auf den Graphen H 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 Q jedesmal an, wenn sie sich ändert und zeichne den gefundenen Baum T.

b) Zeichne einen zusammenhängenden Graphen mit 4 Knoten, der keinen Eulerweg, keine Eulertour, keinen Hamiltonpfad und keinen Hamiltonkreis enthält.

c) Beweise: Ein zusammenhängender Graph mit n Knoten hat mindestens $n - 1$ Kanten.

2.Aufgabe: Binäre Suchbäume (keine AVL-Bäume) 8+2+1 Punkte

a) Füge nacheinander die folgenden Elemente in einen zu Beginn leeren binären Suchbaum ein. Gib den Baum nach jeder Einfügeoperation an:

$$2, 15, 5, 18, 9, 4, 3, 5$$

b) Lösche die 2 aus dem konstruierten Baum. Beschreibe kurz, wie du dabei vorgehst und gib den Baum nach dem Löschen an.

c) Ist der enstandene Baum ein AVL-Baum? Begründe Deine Antwort.

3.Aufgabe: Komplexität **3+3+3+5+5 Punkte**

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

a) Zeige oder widerlege: $f \in O(g), g \in \Theta(h) \Rightarrow f \in O(h)$

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

c) Zeige: $4n^6 + 3n^2 - 48 \in \Omega(n^6)$. Gib dazu explizit geeignete Konstanten c und n_0 aus der Definition an und zeige, dass sie die Definition erfüllen.

d) Sortiere die folgenden Funktionsklassen nach Inklusion (Du musst die Antwort nicht begründen). Kennzeichne identische Klassen und beweise deren Identität.

$$O(n^2) \;\; O(\log n) \;\; O(n^n) \;\; O(n!) \;\; O(\log(n^2)) \;\; O(47^n) \;\; O(\sqrt{n}) \;\; O(354)$$

e) Du hast einen Input von n Elementen und einen Algorithmus, der einzelne Elemente bearbeitet. Das kostet $n^2 + 20 \log n$ Operationen pro Element. Alle Elemente müssen bearbeitet werden.

(i) Wie lange dauert das? (Mit und ohne O-Notation.)

(ii) Die Bearbeitung zweier Elemente ist unabhängig voneinander, man kann dies also parallelisieren. Dazu stehen Dir 4 Kerne zur Verfügung. Wie wirkt sich dies auf die Gesamtlaufzeit aus? (Mit und ohne O-Notation.)

(iii) Alternativ kannst Du ein Preprocessing verwenden, dies läuft in $17n^2 - 3.84n + 715$ und reduziert die Bearbeitungszeiten auf $n \log n$ pro Element, ist aber nicht parallelisierbar, die Bearbeitung der Elemente dann auch nicht mehr. Wie wirkt sich dies auf die Gesamtlaufzeit aus? (Mit und ohne O-Notation.)

(iv) Für welche Möglichkeit entscheidest Du Dich (ursprünglicher Algorithmus, (ii) oder (iii)), um die Leistungsfähigkeit Deines Programmes für große Inputmengen zu maximieren? Begründe Deine Wahl.

4.Aufgabe: Rekursionen **4+4+4+5 Punkte**

a) Wie lautet das Mastertheorem aus der Vorlesung?

b) Bestimme mit Hilfe des Mastertheorems das asymptotische Wachstum der Rekursion
$U(n) = 13 \cdot U(\frac{n}{4}) + 27n^3 + 27 \cdot U(\frac{n}{16}) + 13n$.
Bestimme die Werte aller im Mastertheorem auftretenden Parameter.

c) Bestimme mit Hilfe des Mastertheorems das asymptotische Wachstum der Rekursion
$V(n) = 3125 \cdot V(\frac{n}{5}) + 17n^3$.
Bestimme die Werte aller im Mastertheorem auftretenden Parameter.

d) Bestimme mit Hilfe des Mastertheorems das asymptotische Wachstum der Rekursion
$T(n) = 63 \cdot T(\frac{n}{12}) + 36 \cdot T(\frac{n}{8}) + 51n^2 + 3$.
Bestimme die Werte aller im Mastertheorem auftretenden Parameter.

5.Aufgabe: Hashing **11 Punkte**

Wir betrachten ein leeres Array A der Größe 8, d.h. es gibt die Speicherzellen $A[0], A[1], \ldots, A[7]$; in diesem führen wir offenes Hashing mit der folgenden Hashfunktion durch:

$$t(i,x) = (3x + i^3) \bmod 8$$

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:

$$5, 3, 11, 27, 33$$

(Hinweis: Die Schlüssel sollen in der gegebenen Reihenfolge eingefügt werden und der Rechenweg sollte klar erkennbar sein.)

Trage die Elemente in folgendes Array ein:

0	
1	
2	
3	
4	
5	
6	
7	

6.Aufgabe: Sortieren **10 Punkte**

Sortiere die Zahlen im folgenden Array mit dem in der Vorlesung vorgestellten Quicksort.

$$A[1] = 1 \qquad A[2] = 8 \qquad A[3] = 3 \qquad A[4] = 7$$

Das Referenzelement soll dabei wie in der Vorlesung gewählt werden (also je $A[r]$). Gib das Array nach **jeder** Tausch-Operation an. Gib die Zwischenschritte der Quicksort- und Partitionaufrufe an.

7.Aufgabe: Algorithmenentwurf **4 Punkte**

Gegeben sei eine Zeichenkette, codiert als Array $A[1, \ldots, n]$. Gib einen möglichst schnellen Algorithmus an, der überprüft, ob die Zeichenkette ein Palindrom (eine Zeichenkette, die von vorn und von hinten gelesen identisch ist) ist.
Beispielsweise ist $A[1] =$ 'A' $A[2] =$ 'N' $A[3] =$ 'N' $A[4] =$ 'A' ein Palindrom.

8.Aufgabe: Warteschlangen und Stapel **4 Punkte**

Zeige wie eine Warteschlange Q unter der Verwendung zweier Stacks S_1 und S_2 implementiert werden kann. Gib dafür die neuen Operationen ENQUEUE'(Q, x) und DEQUEUE'(Q) an.

9.Aufgabe: Kurzfragen **2+2+2+2+2 Punkte**

a)	In einem Graphen, der Knoten ungeraden Grades enthält, gibt es nie ein Eulerweg.	☐ wahr ☐ falsch
b)	Für einen Stack S mit n Elementen ist POP(S) in $O(n)$.	☐ wahr ☐ falsch
c)	Ein binärer Suchbaum ist höhenbalanciert, wenn sich für jeden inneren Knoten v die Höhe der beiden Kinder von v um höchstens 1 unterscheidet.	☐ wahr ☐ falsch
d)	Ein AVL-Baum wird nach jedem Insert neu rotiert.	☐ wahr ☐ falsch
e)	Die Größe einer Adjazenzliste eines Baumes ist in $O(n)$.	☐ wahr ☐ falsch

Viel Erfolg!!!