Technische Universität Braunschweig
IBR - Abteilung Algorithmik
Prof. Dr. Sándor Fekete
Nils Schweer

Wintersemester 2009/2010

Klausur
Algorithmen und Datenstrukturen
16.02.2010

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 16 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**	Σ
Bearbeitet ($\times$)									
Punkte	**17**	**12**	**12**	**15**	**12**	**12**	**10**	**10**	**100**
Erzielte Punkte									

1.Aufgabe: **Graphen** **3+7+7 Punkte**

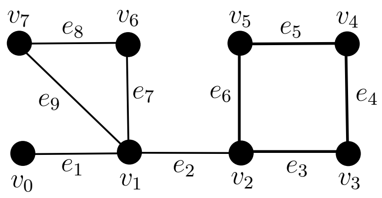

Abbildung 1: Ein Graph.

a) Zeichne einen Graphen mit 4 Knoten und 3 Kanten, der keinen Eulerweg und keinen Hamiltonpfad hat.

b) Wende Fleurys Algorithmus auf den Graphen aus Abbildung 1 an. Starte dabei mit dem Knoten v_0. Können während der Ausführung des Algorithmus in einem Schritt mehrere Kanten ausgewählt werden, wähle die mit dem kleinsten Index.

Gib die Kanten in der Reihenfolge in der sie besucht werden an. Zeichne die gefundene Lösung.

c) Sei $G = (V, E)$ ein ungerichteter Graph und $d(u, v)$ die Länge eines kürzesten Weges zwischen zwei Knoten u und v in G. Der *Durchmesser* von G ist definiert als

$$\max_{u,v \in V} d(u, v) \; ,$$

d.h. als die Länge eines längsten kürzesten Weges.

Gib einen Algorithmus an, der den Durchmesser eines Graphen mit Laufzeit polynomiell in $|E|$ und $|V|$ bestimmt. Beschreibe dazu deinen Algorithmus und begründe seine Laufzeit.

(Hinweis: Ein Korrektheitsbeweis für den Algorithmus muss nicht angegeben zu werden.)

2.Aufgabe: **Binäre Suchbäume** **7+5 Punkte**

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

$$11, 6, 3, 9, 13, 10, 8$$

b) Lösche die 6 aus dem konstruierten Baum. Beschreibe kurz, wie du dabei vorgehst und gib den Baum nach dem Löschen an.

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

Seien $f, g, h : \mathbb{N} \mapsto \mathbb{R}$ drei Funktionen. Sei $f \in \Omega(g)$ und $g \in \Theta(h)$.

a) Zeige oder widerlege: $f \in O(h)$

b) Zeige oder widerlege: $f \in \Theta(h)$

c) Zeige oder widerlege: $f \in \Omega(h)$

d) Zeige: $3n^7 - 4n^3 - 10 \in \Omega(n^6)$. Gib dazu explizit geeignete Konstanten c und n_0 aus der Definition an und zeige, dass sie die Definition erfüllen.

4.Aufgabe: Rekursionen **3+4+4+4 Punkte**

a) Wie lautet das Mastertheorem aus der Vorlesung?

b) Bestimme mit Hilfe des Mastertheorems das asymptotische Wachstum der Rekursion $U(n) = 2 \cdot U(\frac{n}{2}) + 2n^3 + 4 \cdot U(\frac{n}{3})$.

c) Bestimme mit Hilfe des Mastertheorems das asymptotische Wachstum der Rekursion $V(n) = V(\frac{n}{2}) + 4 \cdot V(\frac{n}{8}) + 5n + \log n$.

d) Bestimme mit Hilfe des Mastertheorems das asymptotische Wachstum der Rekursion $T(n) = 4 \cdot T(\frac{n}{2}) + 7n$.

5.Aufgabe: Hashing **12 Punkte**

Wir betrachten ein leeres Array A der Größe 9, d.h. es gibt die Speicherzellen $A[0], A[1], \ldots, A[8]$; in diesem führen wir offenes Hashing mit folgender Hashfunktion durch:

$$t(i, x) = (x + x^2 + i) \bmod 9$$

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:

$$4, 8, 5, 7, 10$$

(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	
8	

6.Aufgabe: Sortieren **4+2+6 Punkte**

a) Wende die Funktion PARTITION(A,1,6) (aus Quicksort) auf folgendes Array an:

$$A[1] = 1 \qquad A[2] = 8 \qquad A[3] = 2 \qquad A[4] = 5 \qquad A[5] = 3 \qquad A[6] = 4$$

Das Referenzelement soll dabei wie in der Vorlesung gewählt werden (also A[6]). Gib das Array nach **jeder** Tausch-Operation an.

b) Wende die Funktion MERGE(A,1,3,6) (aus Mergesort) auf folgendes Array an:

$$A[1] = 7 \qquad A[2] = 5 \qquad A[3] = 9 \qquad A[4] = 4 \qquad A[5] = 3 \qquad A[6] = 8$$

Trage das Ergebnis in folgendes Feld ein:

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

c) Fülle folgende Tabelle mit den zugehörigen Laufzeiten in O-Notation aus:

	Insertionsort	Quicksort	Mergesort
Best Case	———		———
Average Case			———
Worst Case			

(Hinweis: Kästchen, die mit ——— gefüllt sind, müssen nicht ausgefüllt werden.)

7.Aufgabe: Datenstrukturen **4+3+3 Punkte**

a) Stelle die Adjazenzliste und die Inzidenzmatrix zum Graphen G aus Abbildung 2 auf.

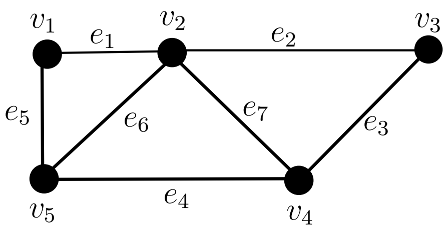

Abbildung 2: Der Graph G.

b) Ergänze die fehlenden Anweisungen in der Operation LIST-SEARCH, die ein Element k in einer doppelt-verketteten Liste L sucht:

LIST-SEARCH(L, k)

x = head[L]

while x $\neq$ . . . . und Wert[x] $\neq$ . . . . **then**

. . . .

end while

return . . . .

c) Wie lange dauert die Suche nach einem Element x im Worst Case in den folgenden Datenstrukturen mit jeweils n Elementen: einfach verkettete Liste, binärer Suchbaum, AVL-Baum.

(Hinweis: Gib deine Aussagen in O-Notation an und skizziere den Worst Case. Trage dabei insbesondere die Position des Elementes x ein.)

8.Aufgabe: Kurzfragen — 2+2+2+2+2 Punkte

a) Ein Stapel arbeitet nach dem LIFO-Prinzip.
☐ wahr
☐ falsch

b) Eine Permutation der Zahlen $1, 2, \ldots, n$ kann in $O(n \log n)$ sortiert werden.
☐ wahr
☐ falsch

c) Das Suchen eines Elementes in einem binären Suchbaum dauert im Worst Case $O(\log n)$.
☐ wahr
☐ falsch

d) Hamiltonpfade benutzen jede Kante höchstens einmal.
☐ wahr
☐ falsch

e) Eulerwege besuchen jeden Knoten höchstens einmal.
☐ wahr
☐ falsch

Viel Erfolg!!!