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

Wintersemester 2010/11

Klausur
Algorithmen und Datenstrukturen
22.02.2011

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 12 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**	**10**	**12**	**16**	**11**	**12**	**12**	**10**	**100**
Erzielte Punkte									

1.Aufgabe: Graphen **8+4+5 Punkte**

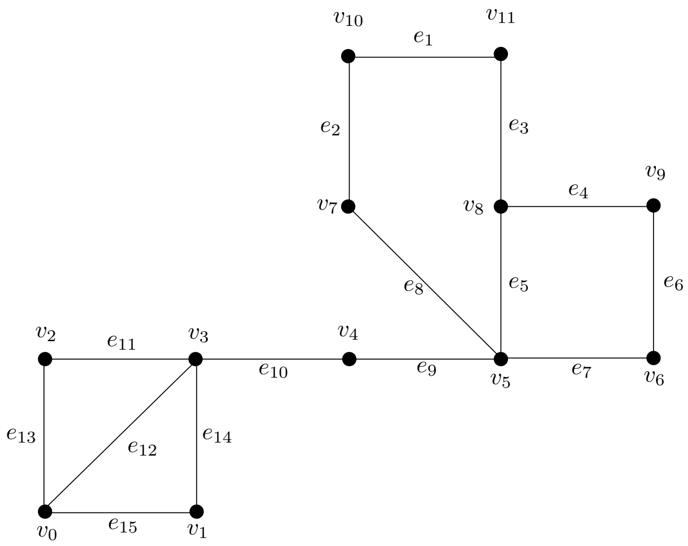

Abbildung 1: Der Graph H.

a) Wende Fleurys Algorithmus aus der Vorlesung auf den Graphen H 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.

b) Zeichnen einen Graphen mit 5 Knoten, der einen Hamiltonkreis, aber keinen Eulerweg hat. Kennzeichne den Hamiltonkreis.

c) Zeige: Die Anzahl der Knoten mit ungeradem Grad in einem Graphen ist immer gerade.

2.Aufgabe: AVL-Bäume **10 Punkte**

Füge nacheinander die folgenden Elemente in einen zu Beginn leeren AVL-Baum ein. Gib den Baum nach jeder Einfüge- und Reparaturoperation (Rotation/Doppelrotation) an:

$$16, 18, 24, 2, 17, 8$$

(Hinweis: Nach *jedem* Einfügen soll der geänderte Baum ein AVL-Baum sein. Zum Schluss sollen alle Zahlen eingefügt sein.)

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

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

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

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

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

d) Zeige: $42n^3 + 17n^2 - 300 \in O(n^4)$. 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+5 Punkte**

a) Wie lautet das Mastertheorem aus der Vorlesung?

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

$U(n) = U(\frac{n}{4}) + 2 \cdot U(\frac{n}{6}) + U(\frac{n}{3}) + U(\frac{n}{12}) + 6n$.

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

$V(n) = V(\frac{n}{3}) + 8 \cdot V(\frac{n}{4}) + 6n^2$.

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

$T(n) = 27 \cdot T(\frac{n}{3}) + 6n^2$.

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:

$$t(i, x) = (x + x^2 + i) \bmod 8$$

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:

$$6, 9, 20, 14, 7$$

(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 **12 Punkte**

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

$$A[1] = 14 \qquad A[2] = 3 \qquad A[3] = 7 \qquad A[4] = 1 \qquad A[5] = 2$$

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: Datenstrukturen **4+4+4 Punkte**

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

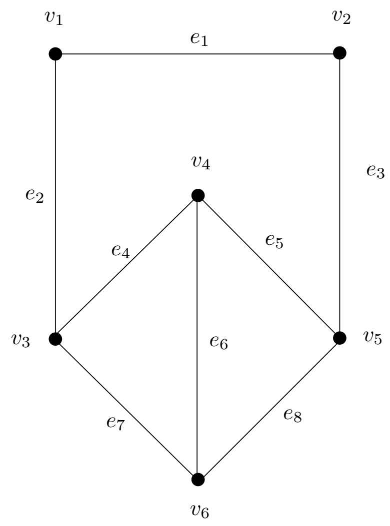

Abbildung 2: Der Graph G.

b) Ergänze die fehlenden Anweisungen in der Operation **Baum-Nachfolger** aus der Vorlesung, die in einem binären Suchbaum den Nachfolger von einem Element i findet.

Baum-Nachfolger(i)

IF r[i] $\neq$ . . . . **THEN**

RETURN . . . .

j:=p[i]

WHILE(j$\neq$ NIL und i=r[j]) **DO**

i:=. . . . .

j:=. . . . .

RETURN j

c) Wie lange dauert das Einfügen eines Elements x sowie die Suche nach einem Element y in den Datenstrukturen AVL-Baum und doppelt verkettete Liste? Gib die Laufzeit in O-Notation und in Abhängigkeit von n an.

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

a) Die Tiefensuche liefert kürzeste Wege in Graphen.
☐ wahr
☐ falsch

b) Verwendet man für die Suche nach einer Zahl x in einer Menge von n sortierten Zahlen die *Binäre Suche*, so benötigt man auch im Worst Case konstante Zeit.
☐ wahr
☐ falsch

c) Ein Stack arbeitet nach dem LIFO-Prinzip.
☐ wahr
☐ falsch

d) Mergesort und Quicksort haben die gleiche Worst-Case-Komplexität.
☐ wahr
☐ falsch

e) Ein Hamiltonpfad benutzt alle Knoten eines Graphen.
☐ wahr
☐ falsch

Viel Erfolg!!!