

Klausur
Algorithmen und Datenstrukturen
23.02.2009

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	9	Σ
Bearbeitet (×)										
Punkte	16	10	10	10	13	12	9	10	10	100
Erzielte Punkte										

1.Aufgabe: Graphen

8+3+5 Punkte

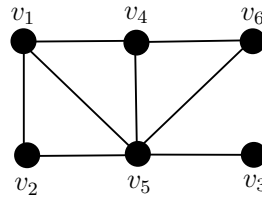


Abbildung 1: Der Graph H .

- 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 .
- Zeichne einen Graphen mit 5 Knoten, der einen Hamiltonpfad aber keinen Eulerweg hat. Kennzeichne den Hamiltonpfad.
- Zeige, dass ein vollständiger Graph mit n Knoten $\frac{n}{2}(n-1)$ Kanten hat.

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:

10, 5, 2, 8, 12, 9

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

3.Aufgabe: Komplexität

5+5 Punkte

- Seien $f, g : \mathbb{N} \mapsto \mathbb{R}$ zwei Funktionen. Zeige:

$$f \in O(g) \Leftrightarrow g \in \Omega(f).$$

- Zeige: $5n^9 + 4n^5 + n \in O(n^{12})$. Gib dazu explizit geeignete Konstanten c und n_0 aus der Definition an und zeige, dass sie die Definition erfüllen.

4.Aufgabe: Rekursionen

4+5+4 Punkte

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

$$U(n) = 2 \cdot U\left(\frac{n}{2}\right) + 2n^2.$$

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

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

c) Zeige mittels vollständiger Induktion:

$$F_n = \frac{(1 + \sqrt{5})^n - (1 - \sqrt{5})^n}{2^n \sqrt{5}}.$$

(Hinweis: Es ist $F_1 = 1$ und $F_2 = 2$)

5.Aufgabe: Hashing

12 Punkte

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

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

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:

5, 14, 10, 23, 16

(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

9 Punkte

Sortiere die folgenden Zahlen mit dem in der Vorlesung vorgestellten Mergesort. Kennzeichne in jedem Schritt, welche Teilfolgen gemischt werden.

[7] [1] [8] [2] [5] [3] [4]

7.Aufgabe: Algorithmenentwurf

10 Punkte

Gegeben sei ein Array A mit $n \geq 2$ positiven Zahlen. Alle Zahlen in dem Array seien unterschiedlich. Gib einen Algorithmus in Pseudocode an, der das **zweitgrößte** Element des Arrays in einer Laufzeit von $O(n)$ ausgibt. (Hinweis: Dein Algorithmus darf nicht mehr als 20 Zeilen haben.)

8.Aufgabe: Datenstrukturen

6+4 Punkte

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

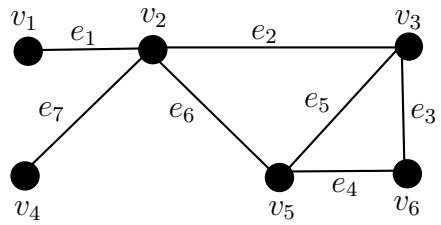


Abbildung 2: Der Graph G .

- b) Ergänze die fehlenden Anweisungen in der Operation LIST-INSERT, die ein Element x an den Anfang einer doppelt-verketteten Liste L einfügt:

LIST-INSERT(L, x)

nachf[x] = head[L]

if head[L] \neq **then**

..... = x

end if

head[L] =

vorg[x] =

9.Aufgabe: Kurzfragen

2+2+2+2+2 Punkte

- a) Eine Warteschlange arbeitet nach dem LIFO-Prinzip. wahr
 falsch
- b) Eine Permutation der Zahlen $1, 2, \dots, n$ kann in $O(\log n)$ sortiert werden. wahr
 falsch
- c) Verwendet man für die Suche nach einer Zahl x in einer Menge von n sortierten Zahlen die *Binäre Suche*, so kann dieses im Worst-Case $\Omega(n)$ dauern. wahr
 falsch
- d) Mergesort hat im Worst-Case die gleiche Laufzeit wie Quicksort (in asymptotischer Notation). wahr
 falsch
- e) Mergesort hat im Average-Case die gleiche Laufzeit wie Quicksort (in asymptotischer Notation). wahr
 falsch

Viel Erfolg!!!