

Vordiplomklausur

Datenstrukturen und Programmierverfahren

Wintersemester 1998/99

Name:
Vorname:
Matrikelnummer:
Studienfach:

Wichtige Hinweise:

1. Prüfen Sie Ihr Klausurexemplar auf Vollständigkeit.
2. Alle 9 Aufgaben sind zu bearbeiten.
3. Es sind keine Hilfsmittel zugelassen.
4. Die Klausur dauert 80 Minuten.
5. Das Deckblatt sowie alle Aufgabenblätter und Lösungsblätter sind abzugeben.

	maximale Anzahl Punkte	erreichte Anzahl Punkte
Aufgabe 1	9	
Aufgabe 2	15	
Aufgabe 3	15	
Aufgabe 4	5	
Aufgabe 5	6	
Aufgabe 6	5	
Aufgabe 7	5	
Aufgabe 8	10	
Aufgabe 9	10	
	80	

1. (3 + 3 + 3 Punkte)

Es soll eine Zahlenfolge A der Länge n aufsteigend sortiert werden. Kreuzen Sie für die verschiedenen Sortierverfahren jeweils die gültigen oberen Schranken für den Sortieraufwand an. Falls Ihrer Meinung nach keine Schranke angegeben werden kann, so kreuzen Sie *keine Angabe möglich* an. Beachten Sie die angegebenen Eigenschaften von A . Gehen Sie von der Laufzeit für den best-case aus. Es werden nur Punkte vergeben, wenn für ein Sortierverfahren die Grenzen korrekt angegeben werden!

(a) Sei A aufsteigend sortiert.

	$O(\lg n)$	$O(n)$	$O(n \lg n)$	$O(n^2)$	keine Angabe möglich
Insertion-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Merge-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Heap-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Quick-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Bucket-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Counting-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Radix-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

(b) Sei A absteigend sortiert.

	$O(\lg n)$	$O(n)$	$O(n \lg n)$	$O(n^2)$	keine Angabe möglich
Insertion-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Merge-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Heap-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Quick-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Bucket-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Counting-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Radix-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

(c) Sei A unsortiert.

	$O(\lg n)$	$O(n)$	$O(n \lg n)$	$O(n^2)$	keine Angabe möglich
Insertion-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Merge-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Heap-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Quick-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Bucket-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Counting-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Radix-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2. (5 + 5 + 5 Punkte)

Seien $f(n)$, $g(n)$ und $h(n)$ asymptotisch positive Funktionen. Beweisen oder widerlegen Sie folgende Aussagen.

- (a) $f(n) = \Theta(g(n)) \wedge g(n) = \Theta(h(n)) \Rightarrow f(n) = \Theta(h(n))$
- (b) $f(n) + \Theta(g(n)) = \Theta(g(n))$
- (c) $f(n) = \Theta(f(n/2))$

3. (5 + 5 + 5 Punkte)

Bestimmen Sie eine asymptotische Lösung (Θ) für folgende Rekurrenzen.

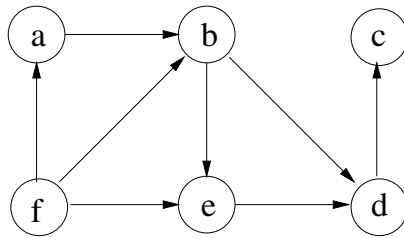
- (a) $T(n) = 3T(n/2) + n$
- (b) $T(n) = 2T(n/4) + \sqrt{n}$
- (c) $T(n) = T(n-1) + 1/n$

4. (5 Punkte)

Reichern Sie einen Rot-Schwarz-Baum mit Attributen an, so daß der arithmetische Mittelwert der enthaltenen Elemente in konstanter Zeit bestimmt werden kann. Die asymptotische Laufzeitkomplexität der Einfüge- und Löschoperation im Rot-Schwarz-Baum soll dadurch nicht beeinflußt werden. Begründen Sie, warum Ihre Lösung diese Bedingung erfüllt.

5. (4 + 2 Punkte)

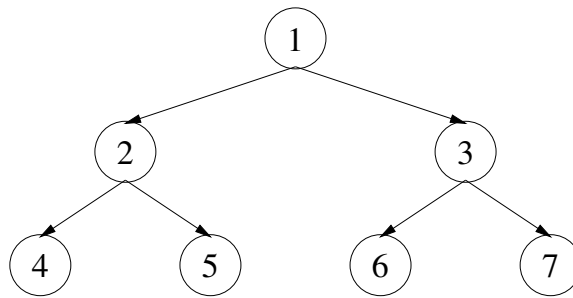
Es sei folgender Graph gegeben.



- (a) Geben Sie einen Breitensuchbaum an, der durch eine im Knoten a beginnende Breitensuche gebildet wird.
- (b) Geben Sie eine topologische Sortierung der Knoten des gegebenen Graphen an.
-

6. (5 Punkte)

Es sei folgender Graph gegeben.



Kreuzen Sie an, ob die gegebene Knotenreihenfolge jeweils einer Reihenfolge entspricht, in der die Knoten bei einer Tiefensuche besucht werden und kreuzen Sie an, ob die gegebene Knotenreihenfolge der Besuchsreihenfolge einer Folge von Breitensuchen entspricht.

	Tiefensuche	Folge von Breitensuchen
1,2,3,4,5,6,7	<input type="checkbox"/>	<input type="checkbox"/>
4,5,6,7,2,3,1	<input type="checkbox"/>	<input type="checkbox"/>
7,6,5,4,3,2,1	<input type="checkbox"/>	<input type="checkbox"/>
2,4,5,1,3,6,7	<input type="checkbox"/>	<input type="checkbox"/>
4,5,2,4,5,6,7	<input type="checkbox"/>	<input type="checkbox"/>

7. (5 Punkte)

Geben Sie einen Tiefensuchalgorithmus in Pseudo-Code an, der mit der Wurzel beginnend, die Knoten eines Binärbaums genau in der Reihenfolge der Endzeiten ausgibt. Seien $\{a_1, \dots, a_n\}$ die Ausgabezeiten der Knoten $\{1, \dots, n\}$ und $\{f_1, \dots, f_n\}$ die Endzeiten der Knoten. Für beliebige Knoten $i, j, i \neq j$ soll gelten:

$$a_i < a_j \Leftrightarrow f_i < f_j$$

8. (10 Punkte)

Sei A eine Zahlenfolge der Länge n . Die Werte von A liegen in $\{1, \dots, n^2\}$. Beschreiben Sie einen Algorithmus der A in $O(n)$ sortiert. Begründen Sie die Korrektheit Ihrer Lösung.

9. (10 Punkte)

Sei $G = (V, E)$ ein gewichteter gerichteter Graph und W eine positive ganzzahlige Konstante. Die Gewichtsfunktion von G sei $w : E \rightarrow \{0, \dots, W - 1\}$. Geben Sie eine Modifikation des Dijkstra-Algorithmus an, so daß die Laufzeit zur Berechnung der kürzesten Pfade in G bezüglich eines Knotens $s \in V$ $O(WV + E)$ beträgt.