

Algorithmen und Datenstrukturen (PI.ADS.AD.VO)	schriftliche Einzelprüfung	27.09.2010		1
--	-------------------------------	------------	--	---

Aufgabe 1 [20]

- [5] Überlegen Sie sich Parameterwerte (a,b und c), die bei Anwendung des Mastertheorems zu einem Gesamtaufwand von $\Theta(n^4 \log n)$ führen.
- [15] Geben Sie in C++ artigem Pseudocode eine rekursive Funktion an, deren Laufzeitordnung $\Theta(n^4 \log n)$ ist.

Aufgabe 2 [20]

Addieren Sie zu Ihrer Matrikelnummer die Zahl 521786349. Die Ziffern der Summe seien in der Reihenfolge von links nach rechts in einem Array gespeichert. Sortieren Sie die Ziffern aufsteigend mit

- [8] Quicksort. Wählen Sie als Pivotelement immer die erste (ganz linke Ziffer).
- [8] Heapsort.
- [4] Mergesort.

Geben Sie jeweils alle benötigten Zwischenschritte so genau an, dass der Ablauf des Algorithmus klar ersichtlich wird.

Aufgabe 3 [15]

- [4] Erstellen Sie einen Expression Tree für den Ausdruck $5/(4-2)*7+(3-6)*(8*9+1)$.
- [9] Geben Sie die drei Reihenfolgen an, in denen die Knoten des Suchbaums bei inorder-, preorder- und postorder-Traversierung bearbeitet werden.
- [2] Was versteht man unter einem perfekten binären Baum?

Aufgabe 4 [25]

- [10] Fügen Sie in eine zu Beginn leere Hastabelle der Größe 7 die Zahlen 8, 4, 1, 11, 5 und 12 nacheinander in dieser Reihenfolge ein. Verwenden Sie als Hashfunktion die Funktion $h(k) = k \bmod 7$. Zur Behandlung von Kollisionen ist double hashing mit der Kollisionsfunktion (zweite Hashfunktion) $g(k) = (k \bmod 10) * 3$ (letzte Stelle von k mal 3) zu verwenden. Skizzieren Sie die resultierende Hashtabelle.
- [5] In der resultierenden Hashtabelle soll der Wert 32 gesucht werden. Geben Sie die Positionen der Tabelle, die überprüft werden müssen, in der Reihenfolge an, in der sie durchsucht werden.
- [5] Warum ist die hier verwendete Kollisionsfunktion (zweite Hashfunktion) nicht empfehlenswert? (Tipp: Betrachten Sie das Ergebnis für $k=20$.)
- [2] Wozu dient die Markierung 'wiederfrei' beim double hashing Verfahren?
- [3] Beschreiben Sie die Kollisionsbehandlungsverfahren separate chaining und linear probing.

Aufgabe 5 [20]

Gegeben ist die folgende Adjazenzmatrix, die die Kosten der Verbindungen zwischen den Knoten eines gerichteten Graphen beschreibt:

$$\begin{pmatrix} 0 & 15 & 3 & 0 & 0 \\ 0 & 0 & 12 & z7 & 0 \\ 0 & z6 & 0 & z5 & 5 \\ z4 & 7 & z3 & 0 & z2 \\ 0 & z1 & 0 & 0 & 0 \end{pmatrix}$$

- [2] Ersetzen Sie in der Adjazenzmatrix die Gewichte $z1$ bis $z7$ durch Werte, die Sie aus Ihrer Matrikelnummer wie folgt ermitteln: zi ergibt sich aus der i -ten Stelle der Matrikelnummer (von rechts beginnend nummeriert) plus 1. Für die Matrikelnummer 1234567 wäre $z2$ beispielsweise 7 ($=6+1$).

Skizzieren Sie den Graphen, der durch diese Adjazenzmatrix beschrieben wird.

- [10] Bestimmen Sie mit dem Algorithmus von Dijkstra die kürzesten Wege vom Knoten 1 zu allen anderen Knoten des Graphen (Dabei entspricht Knoten 1 dem Knoten der ersten Zeile/Spalte in der Adjazenzmatrix).
- [8] Bestimmen Sie mit dem Algorithmus von Prim einen minimal spannenden Baum des Schattens des Graphen. (Sie erhalten den Schatten des Graphen, indem Sie die Richtungen der Kanten vernachlässigen. Werden dann zwei Knoten durch zwei Kanten verbunden, so werden diese Kanten zu einer zusammengefasst. Anders ausgedrückt: Zwei Knoten x und y im Schatten sind genau dann durch eine ungerichtete Kante verbunden, wenn im ursprünglich gerichteten Graphen zumindest eine der Kanten von x nach y oder von y nach x existiert. Als Gewicht der ungerichteten Kante wählen Sie jeweils das Minimum aller durch sie repräsentierten gerichteten Kanten.) Notieren Sie alle Zwischenschritte so genau, dass klar ist, wann welche Kante zum spannenden Baum hinzugefügt wird.