

**Aufgabe 1 Priority Queue****18 Punkte**

In dieser Aufgabe verwenden wir einen Heap als Implementierung einer Priority Queue.

- (a) Fügen Sie die nachfolgenden Zahlen in der gegebenen Reihenfolge in einen Minimum-Heap ein. Zeichnen Sie den Baum nach jeder Einfügeoperation. 12 Punkte  
3, 21, 1, 32, 7, 6, 9, 2
- (b) Löschen Sie aus dem im Aufgabenteil (a) erzeugten Heap dreimal das minimale Element. Geben Sie den Inhalt des Baums nach jedem Löschvorgang an. 6 Punkte

**Aufgabe 2 Priority Queues****20 Punkte**

Bei der Verwendung von Priority Queues werden die Elemente gemäß ihrer Priorität geordnet. Elemente mit gleicher Priorität werden gleich behandelt. Im praktischen Beispiel, z.B. im Wartezimmer eines Arztes, macht es aber neben der Beurteilung der Dringlichkeit eines Falles auch Sinn, Fälle gleicher Priorität unterschiedlich zu behandeln (*first come first serve*).

Erklären Sie, wie eine solche Erweiterung der Datenstruktur für Priority Queues aussieht und geben Sie die veränderten Algorithmen für *insert* und *deletemin* an.

**Aufgabe 3 AVL-Bäume****22 Punkte**

- (a) Zeigen Sie, dass für die Anzahl  $T$  der Knoten eines AVL-Baums der Höhe  $h$  gilt: 17 Punkte

$$\frac{5+2\sqrt{5}}{5} \left( \frac{1+\sqrt{5}}{2} \right)^h + \frac{5-2\sqrt{5}}{5} \left( \frac{1-\sqrt{5}}{2} \right)^h - 1 \leq T \leq 2^{h+1} - 1$$

Hinweis: Es gelten  $\frac{1+\sqrt{5}}{2} + 1 = \left( \frac{1+\sqrt{5}}{2} \right)^2$  und  $\frac{1-\sqrt{5}}{2} + 1 = \left( \frac{1-\sqrt{5}}{2} \right)^2$ .

- (b) Existiert ein AVL-Baum der Höhe 7 mit 50 Schlüsseln? Begründen Sie Ihre Antwort. 5 Punkte

**20 Punkte      Aufgabe 4      Durchläufe von binären Bäumen**

Der Inorder-Durchlauf eines binären Baums liefert folgende Reihenfolge der Knoten:

*A, D, C, G, B, H, E, F, I*

Der Preorder-Durchlauf desselben binären Baums liefert folgende Reihenfolge der Knoten:

*C, D, A, F, G, H, B, E, I*

- 8 Punkte      (a)      Geben Sie die Reihenfolge der Knoten an, die sich bei einem Postorder-Durchlauf durch diesen binären Baum ergeben.
- 12 Punkte      (b)      Geben Sie einen Algorithmus an, der aus der In- und Preorder-Reihenfolge der Knoten eines binären Baums die Postorder-Reihenfolge ermittelt. Sie können dabei einen Algorithmus *Postorder (t: tree of elem): list of elem*, der aus einem binären Baum die Postorder-Reihenfolge berechnet, als gegeben annehmen.

**20 Punkte      Aufgabe 5      Hashing**

Fügen Sie die unten angegebene Folge von Strings in eine Hashtabelle der Größe 10 ein. Verwenden Sie als Hashfunktion die Mittel-Quadrat-Methode und Quadratisches Sondieren als Kollisionsstrategie. Um den initialen Zahlenwert für die Mittel-Quadrat-Methode zu erhalten, summieren Sie die Buchstabencodes (a = A = 1, b = B = 2 usw.) der ersten drei Buchstaben der Zeichenkette auf. Bei einer geraden Anzahl von Ziffern wählen Sie bitte die weiter rechts stehende Zahl als mittlere Zahl aus. Geben Sie für jeden Eintrag den initialen Hashwert sowie ggf. die Folge der Positionen an, bei denen eine Kollision auftritt. Geben Sie letztlich die gefüllte Hashtabelle an.

Cplusplus, Java, Prolog, Python, Cobol, Fortran, Modula, QBasic