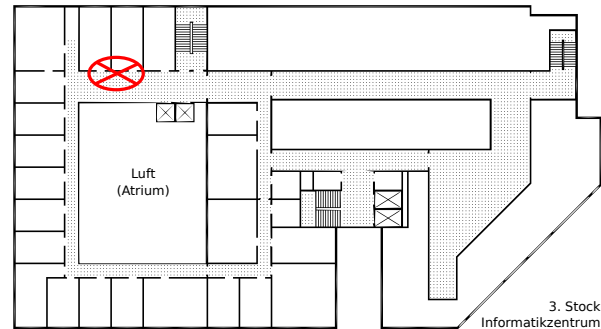


Prof. Dr. Sándor P. Fekete
Dr. Christian Scheffer

Algorithmen und Datenstrukturen Übung 4 vom 9. 1. 2015

Abgabe der Lösungen bis zum Donnerstag,
den 22.01.2015 um 14:30 im Hausaufgabe-
benrückgabeschrank.

Bitte die Blätter zusammenheften
und vorne deutlich mit eigenem Na-
men, Matrikel- und Gruppennummer,
sowie Studiengang versehen!



Aufgabe 1 (Binäre Suchbäume): Wir betrachten binäre Suchbäume (keine AVL-Bäume).

- a) Füge nacheinander die folgenden Elemente in einen zu Beginn leeren binären Suchbaum ein. Gib den Baum nach jeder Einfügeoperation an:

10, 3, 14, 12, 7, 1, 11

- b) Lösche die 14 aus dem konstruierten Baum. Beschreibe kurz, wie du dabei vorgehst. Gib den Baum nach dem Löschen an.
- c) Ist der nach Aufgabenteil b) entstandene Baum ein AVL-Baum? Begründe deine Antwort.

(14+4+2 Punkte)

Aufgabe 2 (Binäre Suchbäume): Wir betrachten einen binären Suchbaum (keinen AVL-Baum) T , der kein Element doppelt enthält.

- a) Gegeben sei ein Knoten v von T mit zwei Kindern. Zeige oder widerlege: Das Minimum im rechten Teilbaum von v hat kein linkes Kind.
- b) Zeige oder widerlege: Wenn T eine Höhe von h hat, dann hat T mindestens $2(h-2)$ Knoten.
- c) Angenommen, die Suche nach einem Element mit Wert k in T endet in einem Blatt. Dieser Suchpfad von der Wurzel zum Blatt teilt die Elemente von T in die drei Mengen A_L , A_P und A_R , die links vom, auf dem, respektive rechts vom Suchpfad liegen.

Zeige oder widerlege: Für alle Elemente $a_1 \in A_L$, $a_2 \in A_P$ und $a_3 \in A_R$ gilt $a_1 \leq a_2 \leq a_3$.

(3+3+4 Punkte)

Aufgabe 3 (AVL-Bäume): Gegeben ist der AVL-Baum T aus Abbildung 1. Führe nacheinander folgende Operationen aus und gib T nach jedem Hinzufügen, Löschen oder Restrukturieren eines Knotens an:

- INSERT(T , 14)
- DELETE(T , 12)
- DELETE(T , 25)
- DELETE(T , 28)
- DELETE(T , 14)

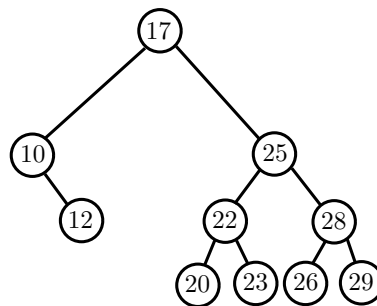


Abbildung 1: Der AVL-Baum

(2+2+2+2+2 Punkte)

Aufgabe 4 (Einfach verkettete Listen): Aus der Vorlesung sind LIST-INSERT und LIST-DELETE für doppelt verkettete Listen bekannt. Wir wollen diese Funktionen für *einfach* verkettete Listen umsetzen.

- Ist LIST-INSERT in $O(1)$ möglich? Wenn ja, wie? Wenn nein, warum nicht?
- Ist LIST-DELETE in $O(1)$ möglich? Wenn ja, wie? Wenn nein, warum nicht?

(Hinweis: Einfach verkettete Listen unterscheiden sich zu doppelt verketteten Listen dadurch, dass die einzelnen Elemente jeweils ihren Schlüssel und Nachfolger, nicht jedoch ihren Vorgänger kennen.)

(5+5 Punkte)

Aufgabe 5 (Stacks und Queues): Eine Warteschlange Q soll unter Verwendung zweier Stacks S_1 und S_2 implementiert werden.

- Beschreibe kurz, wie das funktioniert.
- Definiere für diese Queue ENQUEUE'(Q , x).
- Definiere für diese Queue DEQUEUE'(Q).

(4+3+3 Punkte)