

Algorithmen und Datenstrukturen
SS 2013

Übungsblatt 4: Binäre Suchbäume, O-Notation

Besprechung: 23. - 28.05.2013

Abgabe dieses Übungsblattes bis spätestens Donnerstag, 23.05.2013, 8:00 Uhr.

Hinweise zur Abgabe:

Geben Sie bitte Ihre gesammelten Lösungen zu diesem Übungsblatt in einer Datei `loesung04.zip` unter <https://uniworx.ifi.lmu.de> ab. Bitte verzichten Sie im Dateinamen auf Umlaute, da es sonst zu Problemen mit dem Öffnen des Archivs kommen kann!

Aufgabe 4-1 *Binäre Suchbäume*

4 Punkte

In einem binären Suchbaum sind Zahlen zwischen 1 und 1000 gespeichert. Geben Sie an, ob die folgenden Sequenzen auf der Suche nach der Zahl 333 durchlaufen worden sein können.

- (a) 2, 255, 403, 390, 320, 348, 340, 333
- (b) 783, 153, 720, 245, 733, 300, 340, 333
- (c) 950, 850, 243, 499, 275, 375, 299, 333
- (d) 888, 111, 777, 666, 222, 444, 555, 333

Aufgabe 4-2 *Binäre Suchbäume 2*

4 Punkte

Gegeben sind n Zahlen. Geben Sie mit eigenen Worten ein Verfahren an, um einen binären Suchbaum

- (a) der größtmöglichen Tiefe zu erzeugen.
- (b) der kleinstmöglichen Tiefe zu erzeugen.

Aufgabe 4-3 *Eigenschaften allgemeiner Bäume*

8 Punkte

Wir betrachten einen Baum vom Grad $d \geq 2$.

- (a) Wie groß ist die maximale Anzahl von Knoten auf Level i eines Baumes vom Grad d ?
- (b) Wie groß ist die maximale Anzahl von Knoten in einem Baum der Höhe h vom Grad d ?
- (c) Wie groß ist die maximale Höhe eines Baumes vom Grad d mit n Knoten?
- (d) Wie groß ist die minimale Höhe eines Baumes vom Grad d mit n Knoten?

Begründen Sie Ihre Aussagen.

Aufgabe 4-4 *O-Notation (Fibonaccizahlen)*

8 Punkte

Die Fibonaccifolge ist definiert als: $f_0 = 1, f_1 = 1, f_n = f_{n-1} + f_{n-2}$ (für $n > 1$)

- (a) Geben Sie einen rekursiven Algorithmus zur Berechnung der n -ten Fibonaccizahl in Java-Code an und schätzen Sie dessen Laufzeit mittels O-Notation ab.
- (b) Überlegen Sie, wie dieser Algorithmus effizienter umgesetzt werden kann. Geben Sie auch für diese Umsetzung eine Laufzeitabschätzung in O-Notation an.