Karlsruher Institut für Technologie

Algorithmen 1 – Sommersemester 2018
Zuverlässige Softwaresysteme im Kontext der Automobilindustrie
`http://verialg.iti.kit.edu`

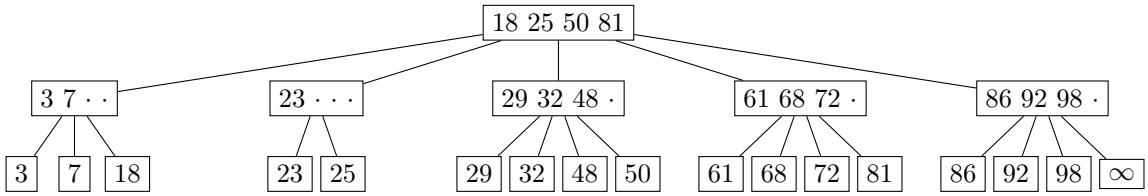
Prof. Dr. Carsten Sinz – Markus Iser

Übungsblatt 8

Ausgabe: 20.06.2018 – 15:30
Abgabe: 27.06.2018 – 13:00

A (a, b)-Bäume

Führen Sie auf dem folgenden $(2, 5)$-Baum die geforderten Operationen in der angegebenen Reihenfolge durch. Zeichnen Sie den Endzustand des Baums nach jeder der angegebenen Operationen.



A.1 Einfügen (1,5 Punkte)

Einfügen von 43, Einfügen von 45, Einfügen von 38.

A.2 Löschen (1,5 Punkte)

Beginnen Sie wieder *beim Ausgangszustand*. Löschen von 50, Löschen von 23, Löschen von 48.

B Binäre Suchbäume

In der Vorlesung wurden Binärbäume mit einer zusätzlichen verketteten Liste an den Blättern vorgestellt. Im Folgenden betrachten wir ausschließlich binäre Suchbäume *ohne zusätzliche Datenliste*. Die Klasse BinaryNode aus Abbildung B für die Knoten ist vorgegeben, wobei left und right auf nil gesetzt werden können.

Wir betrachten in dieser Aufgabe die Liste $A = (42, 13, 57, 91, 45, 48, 6, 51, 10, 55, 13)$.

```
class BinaryNode(key : Element, left, right : BinaryNode) {
  Element key
  BinaryNode left //Alle über left erreichbare Elemente sind <= key
  BinaryNode right //Alle über right erreichbare Elemente sind > key
}
```

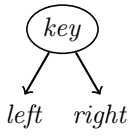


Abbildung 1: Klasse BinaryNode

B.1 Naives Einfügen (2 Punkte)

Fügen Sie schrittweise die Elementen aus Liste A in einen anfangs leeren binären Suchbaum mit der naiven Methode ein. Zeichen den binären Suchbaum nach Einfügen von 45 und 13. Welche Höhe hat der Baum nach Einfügen aller Elemente?

B.2 Von Sortiert zu Balanciert (2 Punkte)

Implementieren Sie einen Algorithmus, der aus einer beliebigen duplikatfreien sortierten Folge L der Länge n einen binären Suchbaum *minimaler Höhe* konstruiert. Begründen Sie kurz warum ein von Ihrem Algorithmus konstruierter Suchbaum diese Eigenschaft haben.

B.3 Ausführung (1 Punkt)

Betrachten Sie anschließend die duplikatfreie, sortierte Liste $L = (6, 10, 13, 42, 45, 48, 51, 55, 57, 91)$ von Elementen aus A. Wenden Sie Ihren Algorithmus aus Teilaufgabe B.2 auf L an, und bestimmen Sie die Höhe des erzielten Baums.

B.4 Find Spezial (2 Punkte)

Implementieren Sie eine Funktion find(), die in einem binären Suchbaum für einen beliebigen Schlüssel k den Knoten x mit dem kleinsten Element $x.key \geq k$ zurückliefert und dabei höchstens h Kanten/Pointer left/right traversiert, wobei h die Höhe des Baums ist.

Karlsruher Institut für Technologie

Algorithmen 1 – Sommersemester 2018
Zuverlässige Softwaresysteme im Kontext der Automobilindustrie
`http://verialg.iti.kit.edu`

Prof. Dr. Carsten Sinz – Markus Iser

Deckblatt Übungsblatt 8

Tutoriumsnummer:

Name	Matrikelnummer	Unterschrift

Mit unseren Unterschriften bestätigen wir, dass wir die Aufgaben eigenständig gelöst haben.

Hinweis: Das Übungsblatt darf in Gruppen von bis zu zwei Personen bearbeitet werden. Beide Personen müssen demselben Tutorium zugeteilt sein. Möchte jemand seine Abgaben-Gruppe innerhalb des Semesters wechseln, so ist dies im Voraus mit dem Tutor abzusprechen. **Bitte tragen Sie oben *groß* die Nummer Ihres Tutoriums ein.** Die Lösung des Übungsblattes ist in jedem Fall mit diesem Deckblatt abzugeben.

Bewertung (durch Tutor):