Prof. Dr. Gerd Stumme, Folke Eisterlehner, Dominik Benz
Fachgebiet Wissensverarbeitung

23/25.06.2009

10. Präsenzübung „Algorithmen und Datenstrukturen“

Sommersemester 2009

Suchbäume

Aufgabe 1 (Einfügen & Löschen in Suchbäumen)

In einen binären Suchbaum sollen nacheinander die folgenden Schlüssel der Folge F eingefügt werden:

$$F := 7, 3, 2, 10, 8, 15, 1, 5, 12, 17$$

a) Geben Sie den Suchbaum an, der dadurch entsteht. Halten Sie sich beim Einfügen neuer Schlüssel strikt an das in der Vorlesung vorgestellte Verfahren; notieren Sie sich für jeden eingefügten Schlüssel die Anzahl der Schlüsselvergleiche, die zum Finden der Einfügeposition notwendig sind.

b) Löschen Sie anschliessend den Schlüssel 10 aus dem Suchbaum. Geben Sie gemäß dem in der Vorlesung vorgestellten Verfahren detailliert alle einzelnen Schritte an, die bei diesem Löschvorgang ausgeführt werden.

Aufgabe 2 (Methode isSearchTree())

Definieren Sie eine rekursive Methode **`isSearchTree(TreeNode node)`**, die den Wert **`true`** zurückliefert, falls der übergebene Binärbaum mit Wurzel **`node`** ein Suchbaum ist; andernfalls soll **`false`** zurückgebeben werden.

Aufgabe 3 (Rekursionsformel für innere Pfadlänge)

$$P(T) := \sum_{v \in T} (Tiefe(v) + 1)$$

bezeichnet mal als die *innere Pfadlänge* eines Suchbaumes T. Sie stellt die Gesamtanzahl der Schlüsselvergleiche dar, wenn nach jedem Schlüssel genau einmal gesucht wird. Stellen Sie eine rekursive Berechnungsvorschrift für die innere Pfadlänge auf.

Aufgabe 4 (Sortieren mit Suchbäumen)

Beschreiben Sie, wie man binäre Suchbäume zum Sortieren benutzen kann. Ein binärer Suchbaum soll durch Einfügen von Elementen aus einer Liste neu aufgebaut werden. Welche Best-/Worst-Case Komplexität hat dieses Verfahren, abhängig von der Länge n der zu sortierenden Folge? Wie sehen die Eingaben aus, die jeweils zur Best-/Worst-Case-Laufzeit führen?

Beschreiben Sie ein Anwendungsszenario, in dem die Verwendung von Suchbäumen zum Sortieren günstig ist.