
5. Übung Algorithmen und Datenstrukturen WS 2016/17

Klaus Kriegel

Abgabe: 25.11.2016, 12:00 Uhr

Aufgabe 1: Traversierung von Binärbäumen (2 + 3 + 2 + 4 Punkte)

Unter Resources/Java finden Sie zwei Klassen für die Knoten von Binärbäumen und für Binärbäume selbst. Das Erzeugen von Beispielen ist etwas beschwerlich, weil man entweder die einzelnen Knoten zuerst mit dem Default-Konstruktor erzeugen und dann alle Referenzen setzen muss oder (etwas komfortabler) den aktuellen Baum unter einem Blatt erweitert und nur die Einträge der neuen Blätter ergänzen muss. Wir wollen hier Bäume mit Einträgen vom Typ `Integer` betrachten und vereinbaren die folgende rekursive Notation: Eine Zahl a ohne Klammern dahinter repräsentiert ein einzelnes Blatt mit den Eintrag a und eine Zahl a mit einem Klammerpaar, welches zwei durch Komma getrennte Repräsentationen von Bäumen b_1, b_2 einschließt, repräsentiert einen Baum, dessen Wurzel den Eintrag a hat mit dem linken Teilbaum b_1 und dem rechten Teilbaum b_2 .

- a) Schreiben Sie ein Testprogramm zur Erzeugung von $8(5(9, 3), 3(6, 2(5, 1)))$.
- b) Implementieren Sie eine Methode `outputBT(BinTree<Integer> t)`, die einen Binärbaum mit `Integer`-Einträgen in der oben vereinbarten Notation ausgibt.
Hinweis: Zur rekursiven Definition ist es sinnvoll, die Arbeit an eine Hilfsmethode auf `BTNode<Integer>` auszulagern.
- c) Implementieren Sie Methoden `depth(BinTree<Integer> t)` und `height(BinTree<Integer> t)`, welche die Einträge der Knoten durch die Tiefen bzw. die Höhen der Knoten ersetzen.
- d) Implementieren Sie die Methoden `List<E> preOrder(BinTree<E> t)` und `List<E> postOrder(BinTree<E> t)`, welche die Einträge der Knoten in der Traversierungsreihenfolge in eine Liste schreiben. Testen Sie die Methoden am Beispiel aus a).

Aufgabe 2: Entscheidungsäume (6 Punkte)

Konstruieren Sie den Entscheidungsbaum, der sich aus der Ausführung des Algorithmus Merge-Sort auf eine Eingabefolge a, b, c, d ergibt. Um es zu vereinfachen, muss man die Teilalgorithmen zum Mischen nur in den Zweigen links von der Wurzel ausführen.

Aufgabe 3: Sortieralgorithmen (5 Punkte)

Wir betrachten Punkte $P = (x, y, z)$ aus dem $n \times n \times n$ -Gitter $G_n = \{0, 1, \dots, n-1\}^3$ mit den Schlüsseln a) $key_1(P) = x + y \cdot n + z \cdot n^2$ bzw. b) $key_2(P) = x + y + z$.

Diskutieren Sie jeweils die (worst case) Laufzeiten der Algorithmen Insertion-Sort, Merge-Sort, Counting-Sort und Radix-Sort zum Sortieren einer Folge von $m = n^2$ Punkten aus G_n (bei bestmöglicher Wahl der freien Parameter in den beiden letztgenannten Verfahren). Alle Laufzeiten sind in möglichst einfacher Form als $\Theta(f(m))$ anzugeben, so dass man eine Reihung der Algorithmen nach Laufzeit erkennen kann.