

8. Übung zur Vorlesung:

Algorithmen und Datenstrukturen

Sommersemester 2009

26. Juni 2009

Abgabe: bis Montag, 06.07.2009, 10:00 h, vor H 426

Aufgabe 8.1: (Punkte 5)

Überlegen Sie sich, wie man einen binären Suchbaum zur Lösung des folgenden Problems verwenden kann: In einer Datei ist ein Text (also eine Folge von Wörtern) abgespeichert. Sie sollen für jedes vorkommende Wort berechnen, wie oft es im Text vorkommt.

Aufgabe 8.2: (Punkte 5)

Implementieren Sie die Rotationsoperationen *rotate_left(v)*, *rotate_right(v)*, *double_rotate_left(v)* und *double_rotate_right(v)* durch entsprechende Funktionen. Nehmen Sie dazu an, dass jeder Knoten v einen Verweis auf seinen Vater-Knoten ($v.parent$) enthält.

Aufgabe 8.3: (Punkte 5)

Betrachten Sie folgende gewichtsbalancierte Balancierungsregel für blatt-orientierte binäre Suchbäume. Für jeden inneren Knoten v gilt:

$$0.5 \leq \frac{\text{gewicht}(T_\ell(v))}{\text{gewicht}(T_r(v))} \leq 2$$

wobei $T_\ell(v)$ der linke Unterbaum von v , $T_r(v)$ der rechte Unterbaum von v , und $\text{gewicht}(T)$ die Anzahl der Blätter im Unterbaum T ist. Zeigen Sie, dass für alle Bäume T mit dieser Eigenschaft $\text{Höhe}(T) = O(\log n)$ gilt.

Aufgabe 8.4: (Punkte 5)

Fügen Sie in einen anfangs leeren Baum aus Aufgabe 8.3 nacheinander die Zahlen von 1 bis 10 ein. Zeigen Sie, wie man durch Rotationen entlang des Suchpfades die Balancierungsbedingung nach jeder Einfügung wiederherstellen kann, falls sie verletzt wurde.