

Algorithmen und Programmierung III

Abgabe am 19. Dezember 2013 vor der Vorlesung in die jeweiligen Tutorenfächer

Aufgabe 1 (2, 3)-Bäume

10 Punkte

- (a) Fügen Sie die Schlüssel A, L, P, D, R, E, I, X, Y, Z in dieser Reihenfolge in einen anfangs leeren (2, 3)-Baum ein. Zeigen Sie die einzelnen Schritte. Löschen Sie sodann die Schlüssel A, L, X.
- (b) Beschreiben Sie schematisch einen (2, 3)-Baum mit n Schlüsseln, so dass eine Löschoperation $\Omega(\log n)$ Rebalancierungsoperationen benötigt.

Aufgabe 2 (a, b)-Bäume

10 Punkte

- (a) Beschreiben Sie, wie man in einem (a, b)-Baum mit n Schlüsseln die Operation `findRange(k_1, k_2)` implementieren kann, die alle Schlüssel k liefert, für die $k_1 \leq k \leq k_2$ ist. Die Laufzeit soll $O(b \log_a n + s)$ betragen. Dabei ist s die Anzahl der gelieferten Schlüssel.
- (b) Seien T_1 und T_2 zwei (a, b)-Bäume, und sei S_1 die Schlüsselmenge von T_1 und S_2 die Schlüsselmenge von T_2 . Alle Schlüssel in S_1 sind kleiner als alle Schlüssel in S_2 . Beschreiben Sie eine Operation `join`, die aus T_1 und T_2 einen (a, b)-Baum für die Schlüsselmenge $S_1 \cup S_2$ erzeugt. Die Laufzeit sollte $O(b \log_a \max\{|S_1|, |S_2|\})$ betragen.
Hinweis: Betrachten Sie zunächst den Fall, dass T_1 und T_2 die gleiche Höhe haben. Achten Sie darauf, dass hinterher die (a, b)-Baum-Eigenschaften wieder hergestellt werden.

Aufgabe 3 Gieriges Münzwechselln

10 Punkte

Betrachten Sie das Problem des Münzwechsellns an der Supermarktkasse. Es soll ein Betrag von n Cents herausgegeben werden, wobei die minimale Anzahl an Münzen verwendet werden soll. Zur Erinnerung: es gibt Euro-Münzen für 1, 2, 5, 10, 20, 50, 100 und 200 Cent.

- (a) Beschreiben Sie einen gierigen Algorithmus für das Problem, und implementieren Sie ihn in Java. Was ist die Laufzeit des Algorithmus?
- (b) Beweisen Sie, dass Ihr Algorithmus aus (a) eine optimale Lösung liefert.
Hinweis: Machen Sie sich Gedanken über die Struktur einer optimalen Lösung: Wie viele 1-Cent Münzen kann eine optimale Lösung höchstens enthalten? Wie viele 2-Cent Münzen? usw?

- (c) Geben Sie ein Beispiel von Münzwerten, so dass der gierige Algorithmus nicht optimal ist. Dabei sollte es eine 1-Cent Münze geben, so dass sich jeder Betrag auch wechseln lässt.