

Algorithmen und Datenstrukturen (WS 2019)

Aufgabenblatt 12

zu bearbeiten bis: 10.02.20-12.02.20

Aufgabe 12.1 (Balancierte Bäume - Theorie)

Wir betrachten die balancierten *2-3-4-Bäume* aus der Vorlesung.

- Fügen Sie nacheinander die Schlüssel $10, 9, 8, \dots, 2, 1$ in einen (anfänglich leeren) *2-3-4-Baum* ein. Zeichnen Sie mindestens einen neuen Baum für alle Zwischenschritte, bei denen Knoten dazu kommen!
- In einem *2-3-4-Baum* sind alle Blätter auf derselben Ebene, aber auch alle inneren Knoten sind jeweils *vollständig*, das heißt wenn sie 2, 3 oder 4 Werte beinhalten haben sie auch genau 3, 4 oder 5 Nachfolger. Überprüfen Sie, ob das auf Ihren finalen Baum zutrifft und korrigieren Sie ggf. Ihre Lösung.
- Das Löschen von Knoten haben wir in der Vorlesung nicht behandelt, unter anderem weil es viele Sonderfälle gibt und man aufpassen muss die Eigenschaften zu erhalten, die einen *2-3-4-Baum* balanciert machen. Gehen Sie durch Ihren finalen Baum und leiten Sie an Hand der Knoten 1 bis 6 **einfache Fälle** für Löschoperationen ab. Versuchen Sie die Fälle als allgemeine Regeln zu formulieren.
- Wie könnte man den Wert 7 löschen? Denken Sie an das Vorgehen beim Löschen aus normalen Binärbäumen!
- Wieso können 8, 9 und 10 mit keinem dieser Ansätze gelöscht werden? Haben Sie eine Idee was mit dem Baum getan werden kann, um das Problem zu lösen?

Aufgabe 12.2 (HashMap - Praxis)

Implementieren Sie eine `HashMap`, welche ein `Array` (oder `ArrayList`) als *Hash-Table* verwendet um Elemente zu speichern. Ihre Klasse soll dabei den abstrakten Datentyp `Map<K, V>` implementieren (so wie bei `TreeMap`). Verwenden Sie zur Bestimmung der Arrayposition für ein Element die `hashCode` Funktion auf dem Schlüssel (Modulo der Arraygröße, achten Sie aber auf negative Hashcodes!). Zu Kollisionsbehandlung verwenden Sie die Strategie **Verkettung**, mit Hilfe von `List<Map.Entry<K, V>>`, also Listen von **Key-Value Paaren**.

Denken Sie daran Ihre `HashMap` zu vergrößern, bevor sie überbelegt wird, damit die Anzahl der Kollisionen nicht zu viel werden!

- Wie würden nun die passenden Implementierungen von `HashSet` und `HashMultiset` aussehen?