

Prof. Dr. Thomas Wiemann

Algorithmen und Datenstrukturen Übungsblatt 8

Wintersemester 2023/24

Aufgabe 8.1 (Implementierung des Einfügens in Rot-Schwarz-Bäume)

Implementieren Sie eine Klasse `RBTree` in einer Datei `RBTree.java`, die einen Rot-Schwarz-Baum repräsentiert. In diesem Testat beschränken wir uns auf die Implementierung des Einfügens neuer Knoten. Schreiben Sie dazu eine Methode `insert`, die sich an der in der Vorlesung vorgestellten `insertNode`-Methode auf Folie 309 orientiert. Anstelle von Integer-Keys sollen in dieser Aufgabe allerdings beliebige Objekte verwaltet werden können, die das Comparable-Interface implementieren. Um die Verletzungen der Rot-Schwarz-Eigenschaften nach dem Einfügen zu reparieren, implementieren Sie die in der Vorlesung vorgestellten Fälle. Dokumentieren Sie in Ihrer Implementierung, welcher Fall an welcher Stelle im Code behandelt wird.

Zum Testen Ihrer Implementierung schreiben Sie eine weitere Datei `Main.java`, die 15 zufällige Integer-Zahlen generiert und in Ihre RB-Tree-Implementierung einfügt. Um mit dem vorgegebenen Interface kompatibel zu sein, erstellen Sie dazu eine Wrapper-Klasse `IntComparable`, die das Comparable-Interface implementiert.

Aufgabe 8.2 (Plotten der Bäume mit DOT)

In dieser Aufgabe plotten wir die Struktur des RB-Trees nach jedem Schritt. Fügen Sie dazu der Klasse `RBTree` eine Methode `printDOT` hinzu, die einen String als Parameter enthält. Dieser definiert eine Datei, in der die aktuelle Struktur des Baumes im sogenannten DOT-Format abgespeichert wird. Die Dokumentation dieses Formats finden Sie hier: <https://graphviz.org/doc/info/lang.html>. Eine Beispieldatei für einen Rot-Schwarz-Baum in diesem Format wird Ihnen in Moodle zur Verfügung gestellt. DOT Files können mit Graphviz in SVG umgewandelt werden. Das Tool erhalten Sie hier: <https://graphviz.org/download/>. Nach der Installation von Graphviz können Sie dazu folgenden Aufruf verwenden: `dot -Tsvg input.dot > output.svg`. Die so entstandenen SVG-Grafiken können Sie dann in das PDF-Format konvertieren. Unter Linux können Sie dazu z.B. das Commandline-Tool `rsvg-convert` benutzen.

Erweitern Sie Ihr Testprogramm um die Funktionalität, dass nach jedem Insert-Schritt die `printDOT`-Methode mit geeignetem Parameter aufgerufen wird. Erzeugen Sie aus den entstandenen DOT-Files eine PDF-Datei, in dem die Struktur Ihres Baumes nach jedem der 15 Insert-Schritte enthalten ist. Hinweis: das Erstellen des PDFs muss nicht automatisch erfolgen, sie können das PDF mit einer geeigneten Toolchain manuell aus den DOT-Files erzeugen. Erklären Sie im Testat die dazu erforderlichen Schritte und machen Sie anhand des entstandenen PDFs deutlich, dass Ihre Implementierung die in der Vorlesung vorgestellten Fälle beim Einfügen korrekt behandelt.

Abgabe

Ihre Abgabe wird in den Übungen in KW 4 testiert. Die Testate erfolgen in 4er-Gruppen. Zur Abgabe laden Sie dazu **pro Person** bis zum **22.01.2023, 8:00 Uhr** Ihre beiden Java-Dateien über die bereitgestellte Abgabefunktion in Moodle hoch. Verspätete Abgaben werden nicht berücksichtigt!