

## Belegaufgabe 1

# Rot-Schwarz Baum

Stefan Schumacher  
(162659) <stefan@net-tex.de>\*

18. Juni 2003

## Inhaltsverzeichnis

---

<b>1 Rot-Schwarz Baum</b>	<b>1</b>
<b>2 Realisierung und Klassen</b>	<b>3</b>
<b>3 Nutzerbeschreibung</b>	<b>3</b>

## 1 Rot-Schwarz Baum

Ein Rot-Schwarz Baum ist die binäre Darstellung eines (2-4)-Baumes mit zusätzlichen Farbinformationen <sup>1</sup> (rot/schwarz) in jedem Knoten.

Weiterhin gelten folgende Eigenschaften:

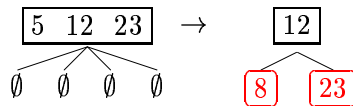
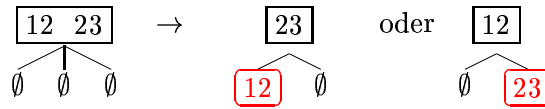
- Wurzel → schwarz
- Blätter → schwarz
- Kinder eines Rotknotens → schwarz
- schwarz → Master
- rot → slave
- 2-Knoten → schwarz
- 3-Knoten → schwarz mit einem roten Kind
- 4-Knoten → schwarz mit zwei roten Kindern
- Schwarztiefe → für alle Blätter gleich

---

\*PGP-Key 0xB3FBAE33

<sup>1</sup>Der Druckbarkeit wegen seien hier eckige Boxen schwarze und abgerundete Boxen rote Knoten

Die Höhe des Baumes ist  $O(\log n)$  und maximal doppelt so hoch wie die des entsprechenden (2-4)-Baumes.



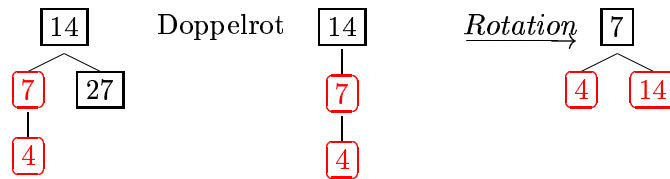
Schwarz-Rot Präsentation eines (2-4)-Baumes

Das Einfügen erfolgt als Blatt mit anschließender Restaurierung der Farbeigenschaften. Wenn der einzufügende Knoten nicht die Wurzel ist, wird er als rotes Blatt eingefügt (Wurzel-, Tiefen- und Blättereigenschaft bleiben dabei erhalten), es kann aber Doppelrot auftreten und es muss restrukturiert werden.

### Restrukturierungsmöglichkeiten

Der Onkel ist schwarz, dies entspricht falschem Einfügen in (2-4)-Baum, siehe Abbildung.

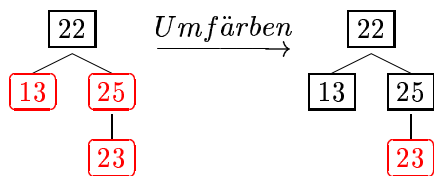
Der mittlere Schlüssel eines 4er Knotens ist schwarz, Rotation durchführen



Rotation im Rot-Schwarz-Baum

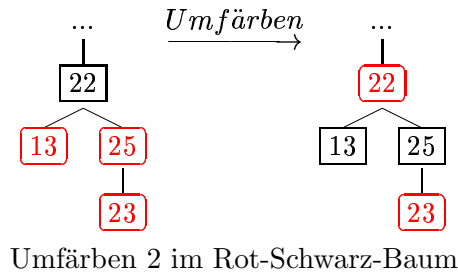
Onkel ist rot, entspricht Schlüsselüberlauf im (2-4)-Baum

Großvater des eingefügten Knoten ist root, kann daher nicht umgefärbt werden. Vater und Onkel werden schwarz umgefärbt, eingefügter Knoten bleibt rot, siehe Abbildung.

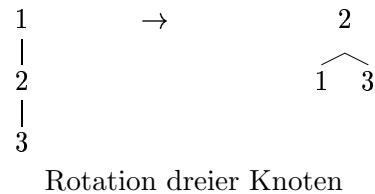


Umfärben 1 im Rot-Schwarz-Baum

Großvater ist nicht `root`, eingefügter Knoten und Großvater sind/werden rot, Vater und Onkel werden schwarz, siehe Abbildung. Umfärbung wird zur Wurzel hin propagiert, daher wächst und verändert sich der Baum von den Blättern nach oben hin.



Neben der Rotation existiert noch die Möglichkeit der Doppelrotation, bei der mit zweimaliger Rotation auch der Fall des Einfügens im linken Teilbaum des rechten Kindes (bzw. umgekehrt) behandelt wird.



## 2 Realisierung und Klassen

Es existieren drei Klassen, `knoten`, `baum` und `gui`. Die `gui`-Klasse realisiert die Oberfläche als Applet, so dass mittels `appletviewer` die Oberfläche genutzt werden kann. Die beiden anderen Klassen realisieren den eigentlichen Baum, wobei `knoten` den Baumknoten implementiert und die Funktionen zum einfärben, Wert setzen und rückgeben sowie vergleichen bereitstellt. Als Objekt werden hier der Einfachheit halber Integer verwendet.

`baum` implementiert die eigentlichen Rot-Schwarz Funktionen wie Einfügen, Rotationen und Färben der Blätter oder Nullen des Baumes. Die Klasse nutzt hierzu eine Statusvariable, die den gegenwärtigen Zustand des Baumes (bspw. unbalanciert oder Wurzel ist rot) beinhaltet und daraufhin die entsprechenden Operationen einleitet.

### 3 Nutzerbeschreibung

Gestartet wird das Applet mittels `appletviewer`, der Nutzer erhält daraufhin die GUI welche 3 Funktionen bietet: ein Textfeld um einen Integer, der als Objekt benutzt wird, zu übergeben, sowie einen Button um den neuen Knoten mit dem entsprechenden Objekt einzufügen und einen Knoten um den Baum zu löschen.

Der erzeugte Baum wird nach dem hinzufügen eines neuen Blattes aktualisiert und graphisch mit entsprechenden Farben angezeigt.

Im unteren Panel befindet sich ein `TextArea`, welches die aufgerufenen Funktionen zum Einfügen des Knoten anzeigt.

Beim Start des Applets wird der aus Aufgabe 79 bekannte Baum aus den Zahlen 1-10 aufgebaut.