

# Bachelorarbeit

Andreas Windorfer

31. März 2020

## Zusammenfassung

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>4</b>
<b>2</b>	<b>Fazit</b>	<b>4</b>
<b>3</b>	<b>Dynamische binäre Suchbäume</b>	<b>4</b>
<b>4</b>	<b>Balancierte Suchbäume</b>	<b>4</b>
4.1	Rot-Schwarz-Baum . . . . .	4
4.1.1	Einfügen beim Rot-Schwarz-Baum . . . . .	6
4.1.2	Löschen beim Rot-Schwarz-Baum . . . . .	7
4.1.3	Suchen beim Rot-Schwarz-Baum . . . . .	7

# 1 Einleitung

# 2 Fazit

# 3 Dynamische binäre Suchbäume

innerer Knoten

Teilbaum

Höhe Nur eine Wurzel vorhanden, dann Höhe 0

innerer Knoten, auch Wurzel

Rotationen

Einfügen

Suchen

Löschen

# 4 Balancierte Suchbäume

## 4.1 Rot-Schwarz-Baum

Der Rot-Schwarz-Baum gehört zur Gruppe der balancierten binären Suchbäume. Zusätzlich zu den Zeigern *links*, *rechts* und *vater* benötigt jeder Knoten ein zusätzliches Bit Speicherplatz, um die Farbinformation zu speichern. Der Name der Datenstruktur kommt daher, dass die beiden durch das zusätzliche Bit unterschiedenen Zustände als *Rot* und *Schwarz* bezeichnet werden. Die Farbe ist also eine Eigenschaft der Knoten. Zeiger auf Kinder oder Väter die aktuell im Baum nicht vorhanden sind zeigen auf einen schwarzen Sonderknoten. Dieser Sonderknoten enthält in dieser Arbeit den Schlüsselwert *null*.

Folgende zusätzliche Eigenschaften müssen bei einem Rot-Schwarz-Baum erfüllt sein.

1. Jeder Knoten ist entweder rot oder schwarz.
2. Die Wurzel ist schwarz.
3. Jedes Blatt ist schwarz.
4. Beide Kinder eines roten Knotens sind schwarz.
5. Für jeden Knoten enthalten alle Pfade, die an diesem Knoten starten und in einem Blatt enden, die gleiche Anzahl an schwarzen Knoten.

Der Zweck dieser Einschränkungen ist es die Höhe des Rot-Schwarz-Baumes zu begrenzen, um die üblichen Operationen effizient ausführen zu können. Als Schwarz-Höhe  $bh(x)$  eines Knoten  $x$  wird die Anzahl der schwarzen Knoten in einem Pfad, der am dem Knotens startet und bei einem Blatt endet, bezeichnet. Die eigene Farbe des betrachteten Knotens bleibt dabei außen vor. Aufgrund der Eigenschaft fünf ist die Schwarz-Höhe eines Knotens wohldefiniert. Die Höhe  $h$  ist beim Rot-Schwarz-Baum wie bei anderen binären Suchbäumen definiert und berücksichtigt auch die roten Knoten.

**Lemma: Maximale Höhe des Rot-Schwarz-Baum:**

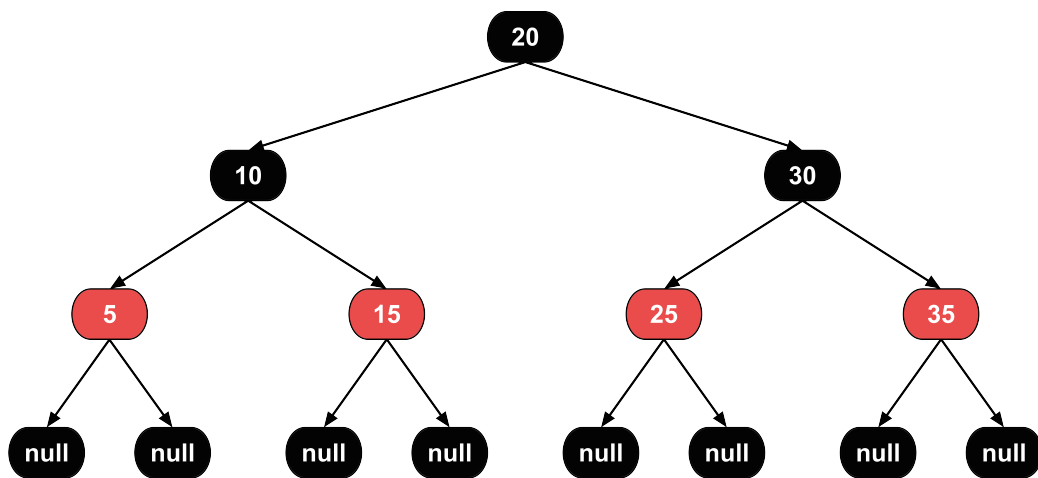


Abbildung 1: Rot-Schwarz-Baum ohne Verletzung von Eigenschaften.

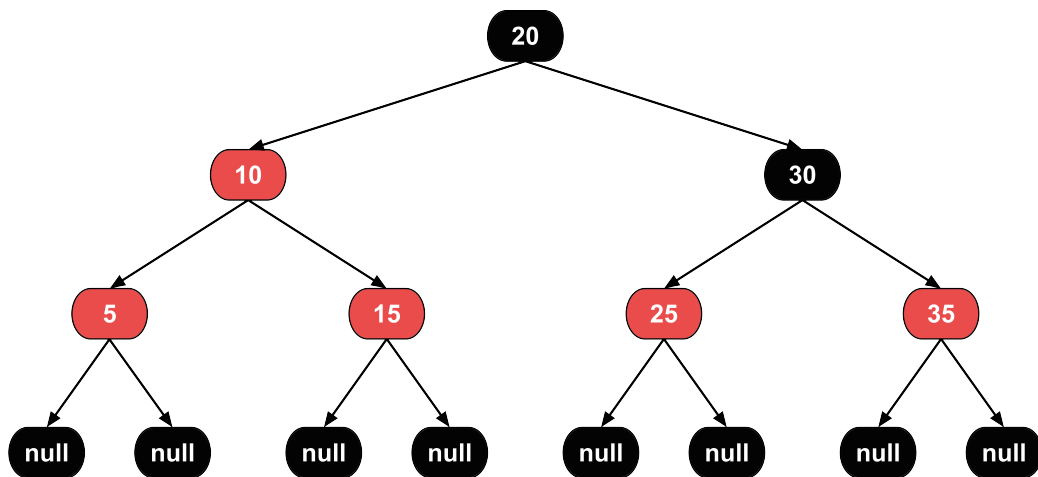


Abbildung 2: Rot-Schwarz-Baum ohne Verletzung von Eigenschaften.

Für einen Rot-Schwarz-Baum mit Höhe  $h$  und  $n$  inneren Knoten gilt  $h \leq$

$2lg(n+1)$ .

**Beweis:**

Zunächst wird gezeigt, dass ein Teilbaum  $t_1$  mit Wurzel  $x_1$  und Schwarzhöhe  $bh(x_1)$  zumindest über  $2^{bh(x_1)} - 1$  innere Knoten verfügt, also  $n \geq 2^{bh(x_1)} - 1$ . Dies wird durch Induktion über die Höhe  $h_{x_1}$  gezeigt. Für  $h_{x_1} = 0$  besteht  $t_1$  nur aus einem Blatt und enthält keine inneren Knoten. Natürlich gilt in diesem Fall auch  $bh(t_1) = 0$ .

Induktionsanfang mit  $h_{x_1} = 0$  :

$$2^0 - 1 = 0$$

Induktionsschritt:

Nun wird ein Teilbaum  $T_2$  mit Wurzel  $x_2$  mit Höhe  $h+1$  betrachtet. Jedes seiner beiden Kinder hat entweder Schwarzhöhe  $bh(x_2)$ , wenn es rot ist, oder  $bh(x_2) - 1$ , wenn es schwarz ist. Die Höhe beider Kinder ist niedriger als die eigene Höhe von  $x_2$ . Somit kann bei beiden Kindern jeweils die Induktionsannahme für die Mindestanzahl der inneren Knoten eingesetzt werden.

$$2^{bh(x_1)-1} - 1 + 2^{bh(x_1)-1} - 1 = 2^{bh(x_1)} - 2$$

Addiert man einen inneren Knoten aufgrund der Wurzel  $x_2$  hinzu, erhält man die Behauptung.

$$2^{bh(x_1)} - 2 + 1 = 2^{bh(x_1)} - 1.$$

Es gilt also  $n \geq 2^{bh(x)} - 1$ .

Auf einem Pfad in einem Rot-Schwarz-Baum  $t_3$  von der Wurzel bis zu einem Blatt sind der erste und der letzte Knoten, sowie mindestens jeder zweite der weiteren Knoten schwarz. Es gilt also  $bh(t_3) \geq \frac{h(t_3)}{2}$ . Damit kann man in der Ungleichung  $bh(t_3)$  durch  $\frac{h(t_3)}{2}$  ersetzen, woraus dann das Lemma folgt.

$$n \geq 2^{\frac{h(t_3)}{2}} - 1 \Rightarrow n + 1 \geq 2^{\frac{h(t_3)}{2}} \Rightarrow \lg(n + 1) \geq \frac{h(t_3)}{2} \Rightarrow 2lg(n + 1) \geq h(t_3)$$

#### 4.1.1 Einfügen beim Rot-Schwarz-Baum

Ein neuer Schlüssel wird zunächst wie in Ref: (normales Einfügen) eingefügt. Zusätzlich werden dann noch die Zeiger auf die Kinder auf den schwarzen Sonderzeiger gesetzt. Der neue Knoten wird rot gefärbt. Durch den neu eingefügten Knoten können Korrekturen notwendig werden, um die Rot-Schwarz-Baum Eigenschaften zu erhalten. Zunächst betrachten wir die fünf Eigenschaften um zu erkennen welche betroffen sein können. Es ist immer noch jeder Knoten entweder rot oder schwarz. Aufgrund des Sonderknotens sind die Blätter immer noch schwarz. Da der neue Knoten rot ist bleiben auch die Schwarz-Höhen der Knoten unverändert, so dass Eigenschaft fünf erhalten bleibt. Wurde in den leeren Baum eingefügt, so wurde jedoch eine rote Wurzel eingefügt. Ist der Vater des neuen Knotens rot, so wurde Eigenschaft vier verletzt. Zur Korrektur wird eine zusätzliche Routine verwendet. Diese arbeitet sich von der Einfügestelle solange nach oben durch, bis alle Eigenschaften

wieder eingehalten werden. Pro Iteration werden acht Fälle unterschieden, von denen jeweils zwei symmetrisch sind.

#### **4.1.2 Löschen beim Rot-Schwarz-Baum**

#### **4.1.3 Suchen beim Rot-Schwarz-Baum**

## Literatur