Sarah Ertel
Patrick Greher
Eugen Ljavin

1	2	3	4	Σ
6	4	6	12	28

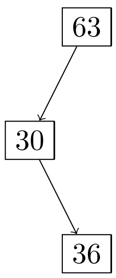
Übungsblatt Nr. 6

(Abgabetermin 07.05.2018)

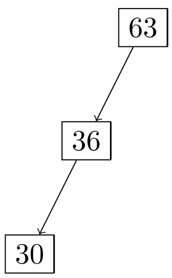
Aufgabe 1

a)

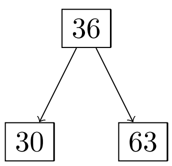
Einfügen von 63, 30, 36:



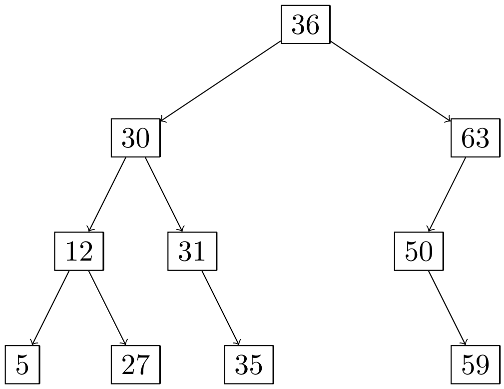
Doppelrotation Links-Rechts zuerst Links:



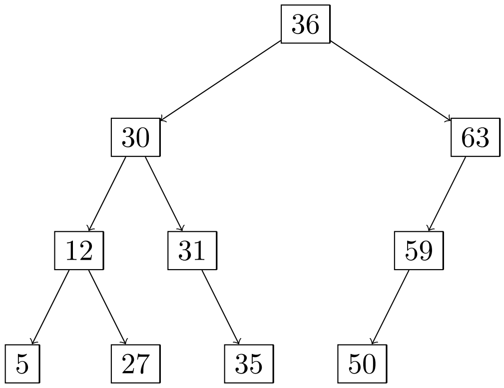
Doppelrotation Links-Rechts dann Rechts:



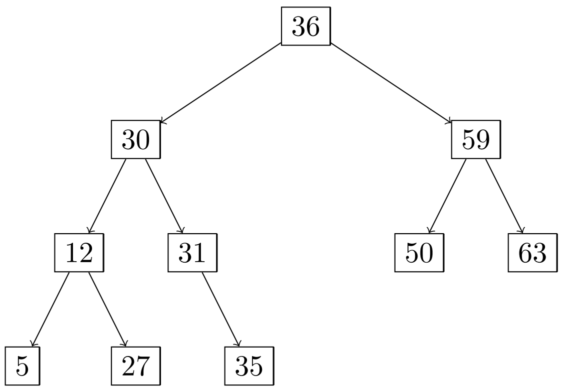
Einfügen von 31, 12, 50, 35, 5, 27, 59:



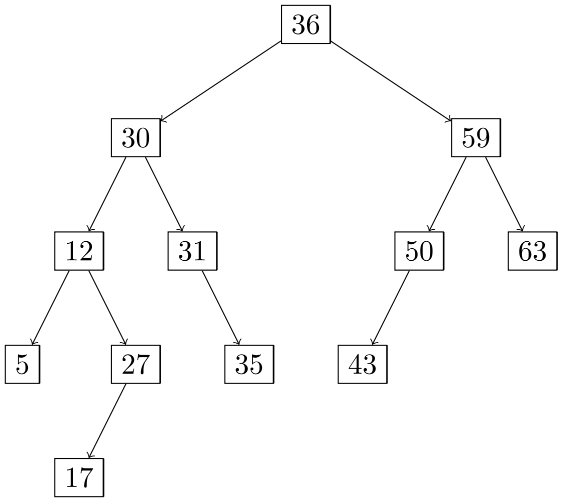
Doppelrotation Links-Rechts zuerst Links:



Doppelrotation Lins-Rechts dann Rechts:



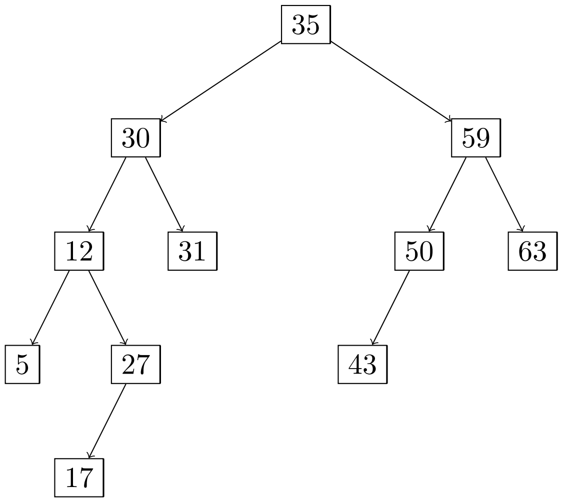
Einfügen von 43, 17:



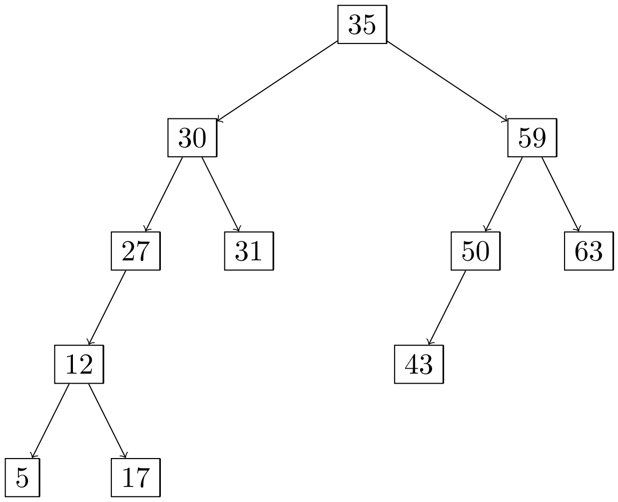
korrekt

b)

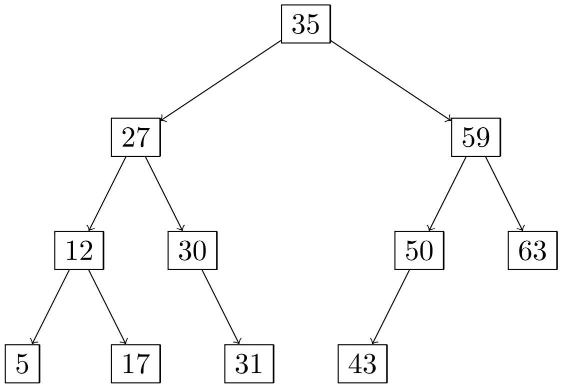
Entfernen des Knotens mit dem Key 36 (root). Der Knoten mit dem Key 35 wird zur neuen root. Rebalancierung erforderlich:



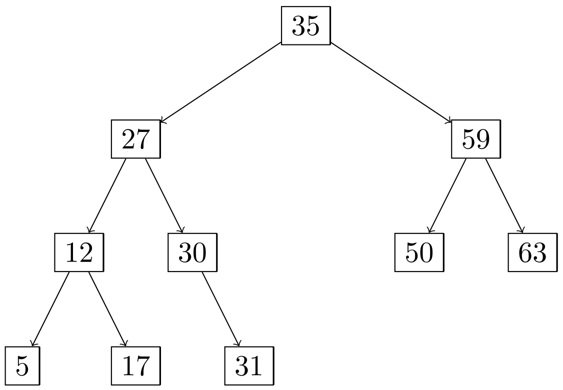
Wiederherstellung der Balance - LR Rotation zuerst Links:



Wiederherstellung der Balance - LR Rotation dann Rechts:



Entfernen des Knotens mit dem Key 43. Keine Rebalancierung erforderlich:



korrekt

c)

Schritt 1: Alle Elemente sortieren
Schritt 2: Median (falls n ungerade) oder abgerundetes Medianelement (falls n gerade) als Wurzel nehmen
Schritt 3: Rekursiv einen linken und rechten Teilbaum erzeugen. In den linken Teilbaum kommen alle Zahlen die kleiner als das Wurzelelement sind, in den rechten Teilbaum alle größeren Zahlen.
Innerhalb der jeweiligen Unterbäume wird dieses Vorgehen wiederholt. D.h. es wird zur Erstellung eines Unterbäumes wiederum der Median gesucht und ein linker und ein rechter Teilbaum erzeugt.

Da beide Teilbäume sich höchstens um einen Knoten unterscheiden, wird der Höhenunterschied ebenfalls maximal 1 sein.
Durch den Rekusiven Aufbau der Teilbäume haben auch die Unterbäume eines Teilbaums höchstens einen Höhenunterschied von 1.

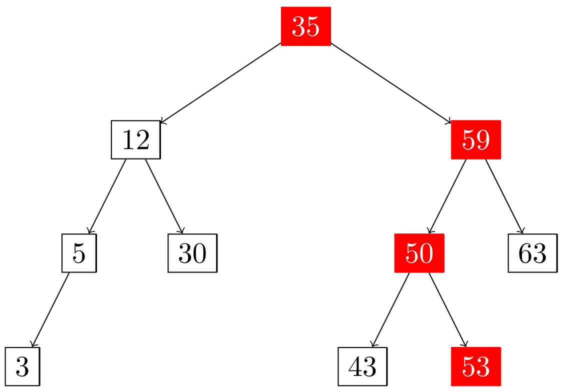
korrekt

Aufgabe 2

a)

Nein die Aussage gilt nicht immer. Es wird beispielsweise der unten abgebildeten binäre Suchbaum mit dem in rot angegebenen Pfad betrachtet. Es gilt $B = \{35, 59, 50, 53\}$, $A = \{12, 5, 3, 30, 43\}$ und $C = \{63\}$ mit $a \in A, b \in B$ und $c \in C$.

Gegenbeweis: Wähle $a = 43$ und $b = 35$. Es gilt $a \nleq b$.



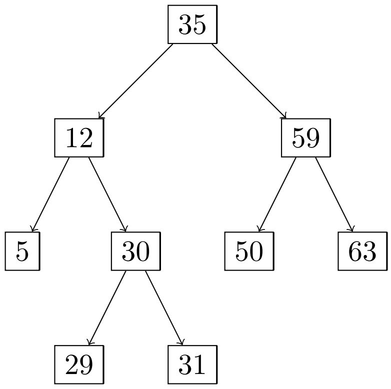
korrekt

b)

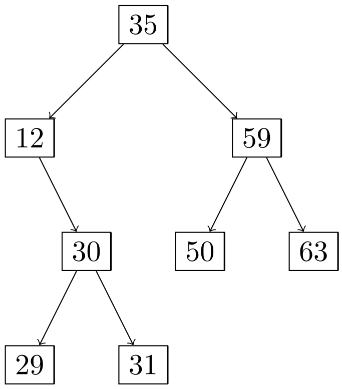
-

c)

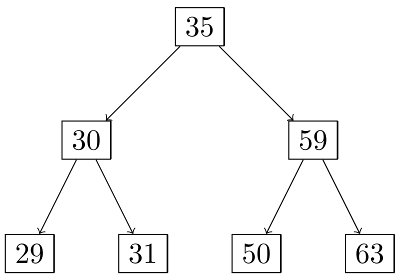
Betrachtet wird der nachfolgende binäre Suchbaum. Im ersten Fall wird zunächst der Knoten mit dem Key 5 gelöscht und anschließend der Knoten mit dem Key 12. Im zweiten Fall wird zuerst der Knoten mit dem Key 12 gelöscht und anschließend der Knoten mit dem Key 5. Da sich die resultierenden binären Suchbäume voneinander unterscheiden, ist das löschen im binären Suchbaum nicht-kommutativ.



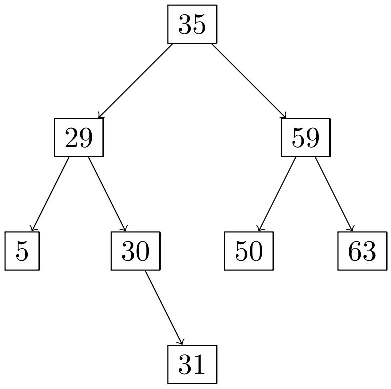
Fall 1: Löschen des Knotens mit dem Key 5:



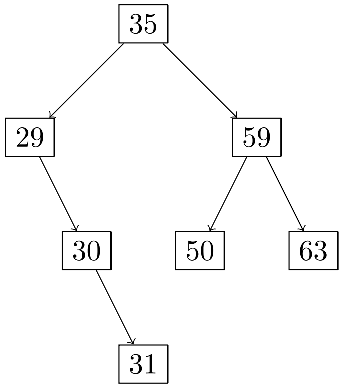
Fall 1: Anschließndes löschen des Knotens mit dem Key 12:



Fall 2: Löschen des Knotens mit dem Key 12:



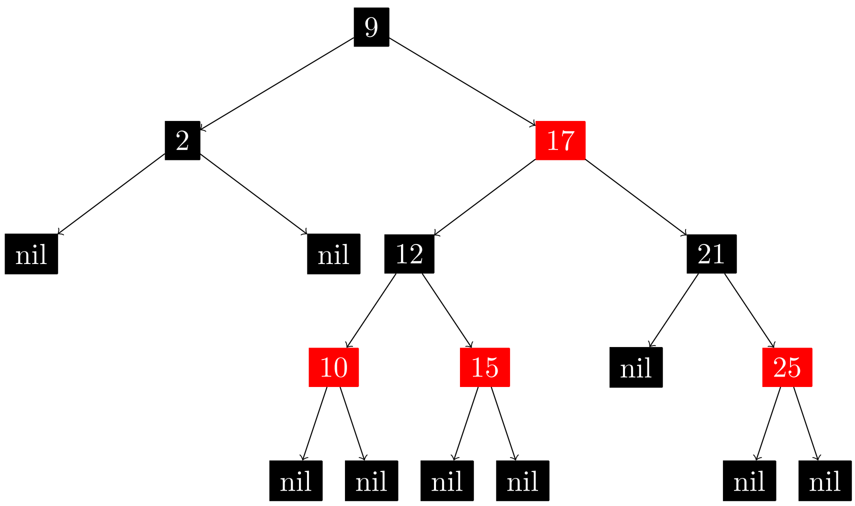
Fall 2: Anschließendes löschen des Knotens mit dem Key 5:



korrekt

Aufgabe 3

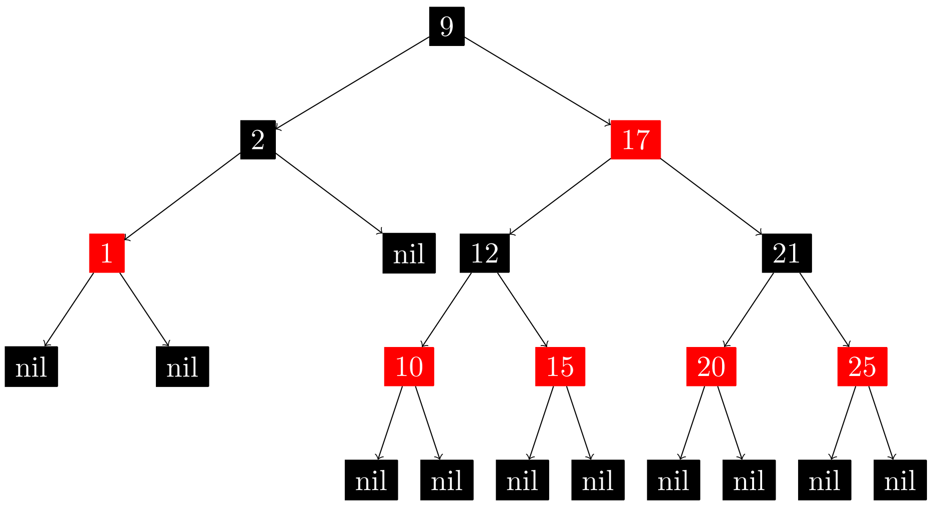
a)



korrekt

b)

Hinzufügen des Knotens mit dem Key 1 und dem Key 20:



korrekt

c)

Die Eigenschaft 5 garantiert, dass ein Rot-Schwarz Baum stets balanciert ist, falls er nur schwarze Knoten enthält. Sei hb die Höhe des Baumes ausgehend von der Wurzel entlang eines Pfades zu einem Blatt, wobei der Baum nur schwarze Knoten enthält. Nach Eigenschaft 5 muss der Baum dann mindestens $2^{hb} - 1$ Knoten haben. Folglich gilt $n \geq 2^{hb} - 1$ und somit aufgelöst $hb = \log_2(n + 1)$.
Die Eigenschaft 4 garantiert, dass keine zwei rote Knoten aufeinanderfolgen können. Zwischen jedem roten Knoten muss somit ein schwarzer Knoten liegen wobei die Anzahl roter Knoten hr entlang eines Pfades ebenfalls höchstens $\log_2(n+1)$ entspricht. Folglich kann es entlang eines Pfades **höchstens** genausoviele rote Knoten wie schwarze Knoten geben. Die maximale Gesamthöhe h eines Pfades entspricht damit der Summe der schwarzen Knoten sowie roter Knoten des Baumes also $h \leq hb + hr$.

$$
\begin{aligned}
n &\geq 2^{hb} - 1 \\
hb &= \log_2(n + 1) \\
hr &\leq \log_2(n + 1) \\
h &\leq hb + hr \\
h &\leq \log_2(n + 1) + \log_2(n + 1) \\
h &\leq 2 \cdot \log_2(n + 1)
\end{aligned}
$$

korrekt

d)

Nein. Bei dem Rot-Schwarz-Baum aus Aufgabenteil a) handelt es sich beispielsweise um keinen AVL Baum, da der Rot-Schwarz-Baum nicht im Sinne eines AVL Baums ausbalanciert ist.

korrekt