Prof. Dr. Ir. Joost-Pieter Katoen
Christian Dehnert, Friedrich Gretz, Benjamin Kaminski, Thomas Ströder

Tutoraufgabe 1 (Vollständige Induktion):

Finden Sie eine geschlossene Form für die Summe

$$\sum_{i=0}^{n} 2^i$$

und beweisen Sie mittels eines geeigneten Verfahrens die Korrektheit der geschlossenen Form.

Lösung:

Für $n = 0$ ergibt sich 1, für $n = 1$ ergibt sich 3, für $n = 2$ ergibt sich 7, usw. Insgesamt schließen wir auf $2^{n+1} - 1$ als Hypothese für eine geschlossene Form der Summenformel. Wir beweisen die Richtigkeit der Hypothese im Folgenden mittels vollständiger Induktion über n:

Induktionsanfang: Für $n = 0$ ergibt sich

$$\sum_{i=0}^{0} 2^i = 2^0 = 1 = 2^{0+1} - 1 \ .$$

Damit stimmt die Aussage für $n = 0$.

Induktionshypothese (IH): Für ein beliebiges aber festes n gilt:

$$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1 \ .$$

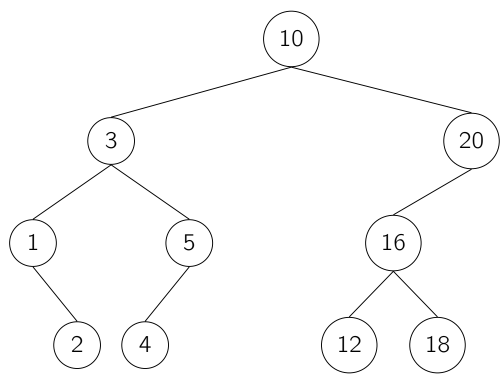
Induktionsschritt: Wir zeigen nun, dass die Aussage unter Annahme der Induktionshypothese auch für $n + 1$ gilt:

$$\sum_{i=0}^{n+1} 2^i = \sum_{i=0}^{n} \left(2^i \right) + 2^{n+1} \overset{\text{IH}}{=} 2^{n+1} - 1 + 2^{n+1} = 2 \cdot 2^{n+1} - 1 = 2^{(n+1)+1} - 1$$

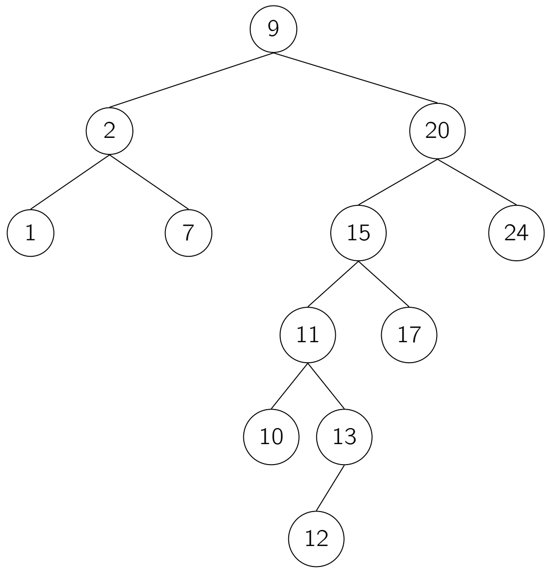
Damit die ist Aussage für $n + 1$ gezeigt. Nach dem Prinzip der vollständigen Induktion gilt die Aussagen damit für alle $n \in \mathbb{N}$.

Tutoraufgabe 2 (Rotationen):

a) Geben Sie für den folgenden binären Suchbaum die Binärbäume an, die entstehen, wenn eine **Rechtsrotation** auf den Knoten mit dem Wert **20** ausgeführt wird, dann eine **Linksrotation** auf den Knoten mit Wert **3** und zuletzt eine **Linksrotation auf die Wurzel** des entstandenen Baumes:

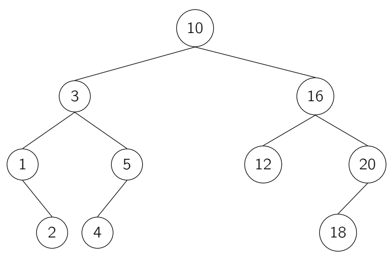
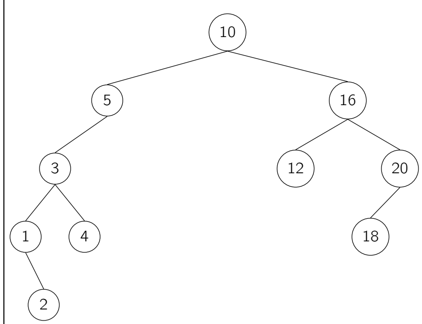
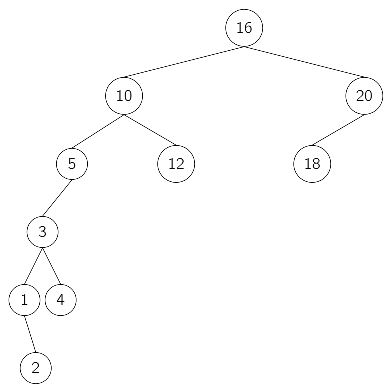


b) Geben Sie **maximal vier Rotationen** an, die den folgenden Baum der **Höhe fünf** in einen Baum der **Höhe drei** transformieren. Geben Sie auch den Zustand des Baumes nach jeder Rotation an.

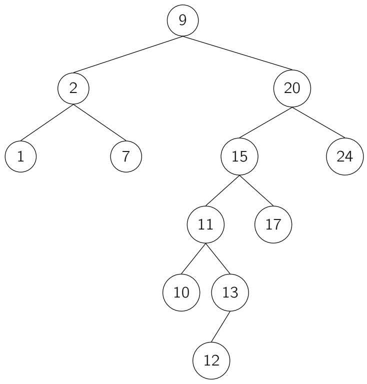
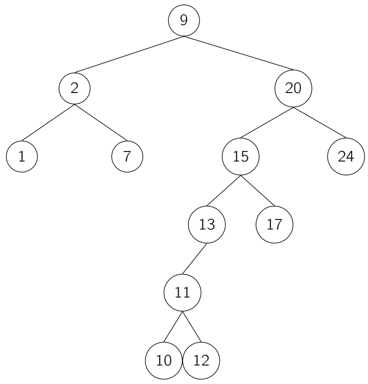
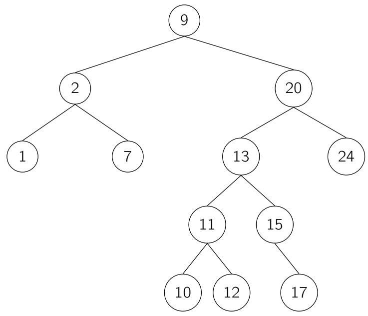


Lösung:

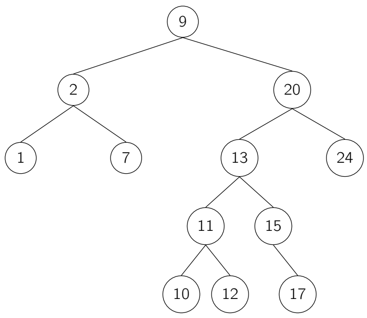
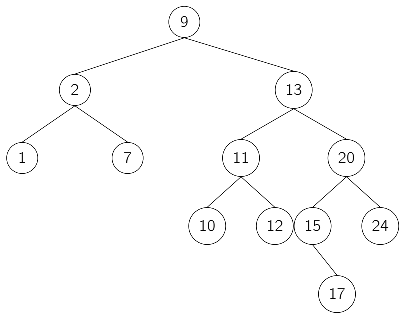
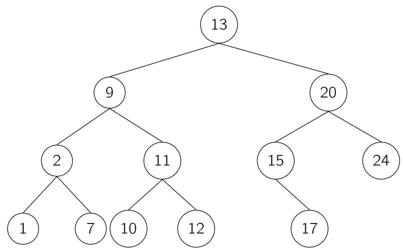
a) Es ergeben sich folgende Bäume durch Rotation:

Rechtsrotation auf Knoten 20:	Linksrotation auf Knoten 3:	Linksrotation auf Wurzel:
		

b) Wir können den gegebenen Baum mit vier Rotationen in einen Baum der Höhe drei transformieren. Hierzu führen wir zuerst eine Linksrotation auf den Knoten 11 aus und anschließend eine Rechtsrotation auf den Knoten 15. Durch diese beiden Rotationen erhalten wir einen Baum der Höhe vier:

Ursprünglicher Baum:	Linksrotation auf Knoten 11:	Rechtsrotation auf Knoten 15
		

Auf diesen Baum führen wir nun noch eine Rechtsrotation auf den Knoten 20 durch, gefolgt von einer Linksrotation um die Wurzel und erhalten so einen Baum der Höhe drei:

Baum der Höhe vier:	Rechtsrotation auf Knoten 20:	Linksrotation auf Wurzel:
		

Tutoraufgabe 3 (AVL Bäume):

Führen Sie folgenden Operationen beginnend mit einem anfangs leeren *AVL-Baum* aus und geben Sie die entstehenden Bäume nach jeder *Einfüge-* und *Löschoperation* sowie jeder *Rotation* an. Markieren Sie außerdem zu jeder Rotation, welcher Knoten in welche Richtung rotiert wird:

1. 9 einfügen

2. 7 einfügen

3. 8 einfügen

4. 3 einfügen

5. 6 einfügen

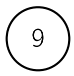
6. 2 einfügen

7. 2 löschen

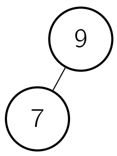
8. 3 löschen

Lösung:

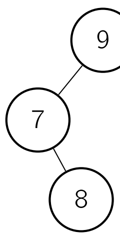
füge 9 ein



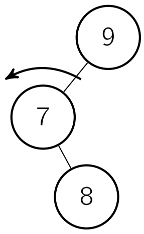
füge 7 ein

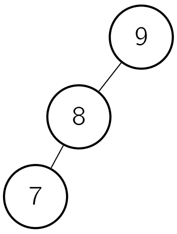


füge 8 ein

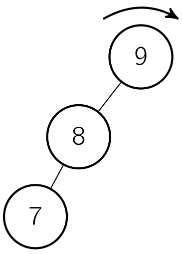


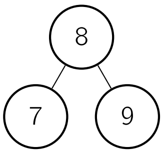
rotiere 7 nach links



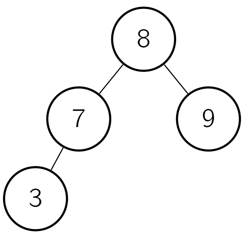


rotiere 9 nach rechts

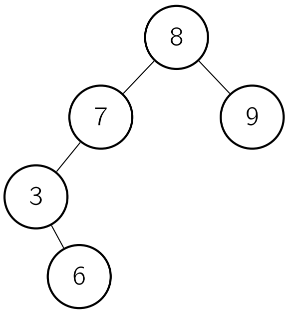




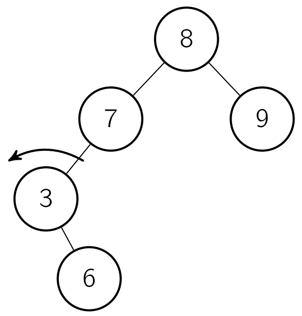
füge 3 ein

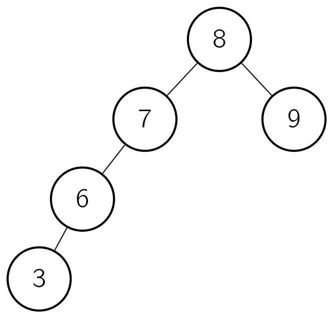


füge 6 ein

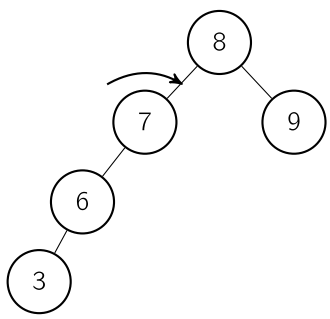


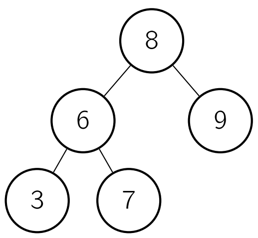
rotiere 3 nach links



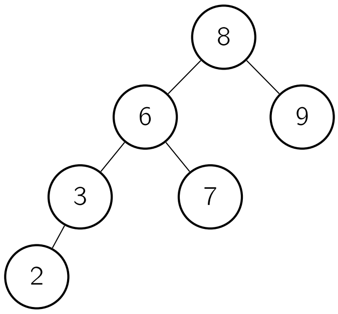


rotiere 7 nach rechts

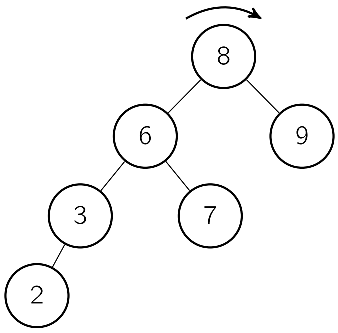


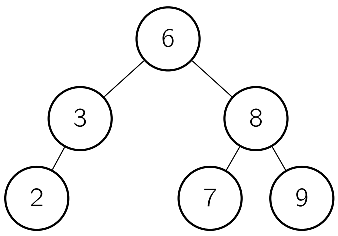


füge 2 ein

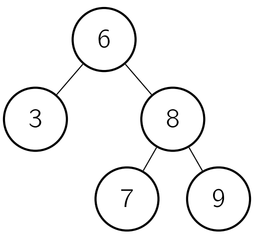


rotiere 8 nach rechts

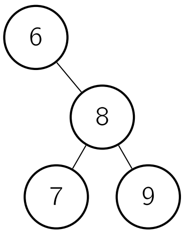




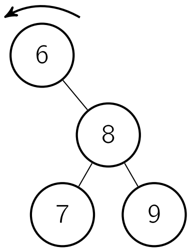
entferne 2

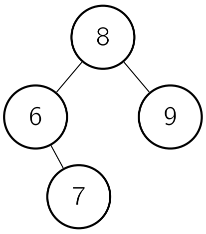


entferne 3



rotiere 6 nach links





Tutoraufgabe 4 (Rot-Schwarz Bäume):

Führen Sie die folgenden Operationen beginnend mit einem anfangs leeren *Rot-Schwarz-Baum* aus und geben Sie die entstehenden Bäume nach

- jeder *Einfügeoperation*,
- jeder *Löschoperation*,
- jeder *Rotation* sowie
- jeder *Umfärbung* an.

Markieren Sie außerdem zu jeder Rotation, welcher Knoten in welche Richtung rotiert wird. Mehrere Umfärbungen können Sie in einem Schritt zusammenfassen. Beachten Sie, dass rote Knoten rund und schwarze Knoten eckig dargestellt werden.

1. 9 einfügen

2. 7 einfügen

3. 8 einfügen

4. 3 einfügen

5. 6 einfügen

6. 2 einfügen

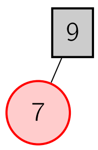
7. 6 löschen

Lösung:

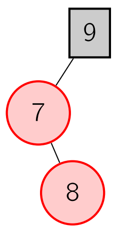
füge 9 ein

9

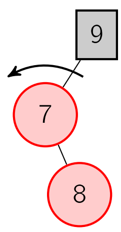
füge 7 ein

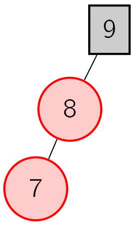


füge 8 ein

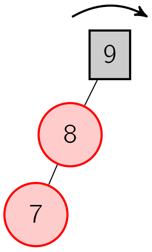


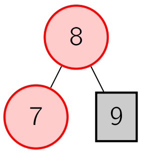
rotiere 7 nach links



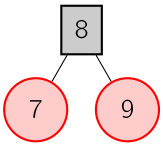


rotiere 9 nach rechts

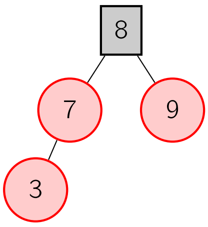




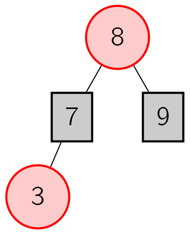
Fall 3: umfärben



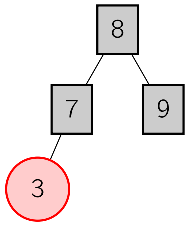
füge 3 ein



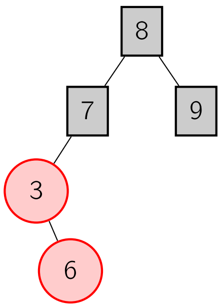
Fall 1: umfärben



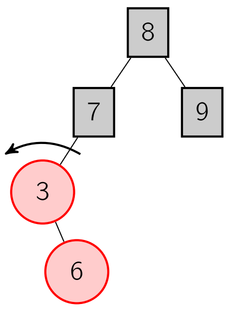
Wurzel schwarz färben

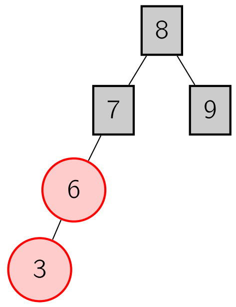


füge 6 ein

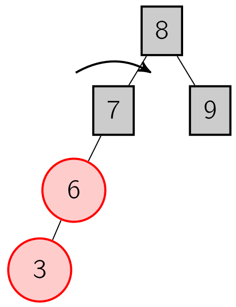


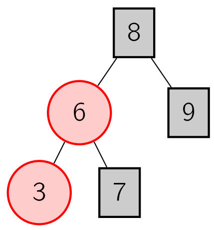
rotiere 3 nach links



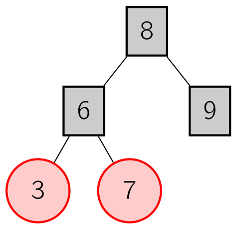


rotiere 7 nach rechts

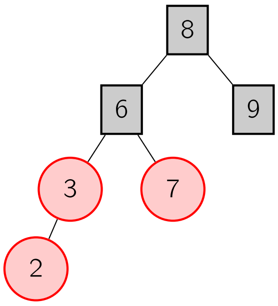




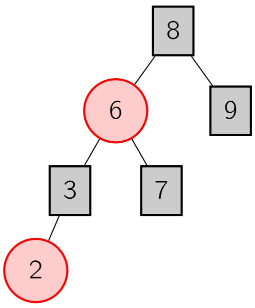
Fall 3: umfärben



füge 2 ein

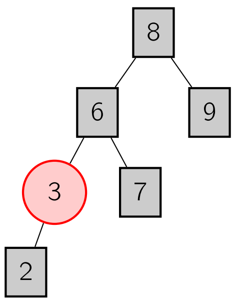


Fall 1: umfärben

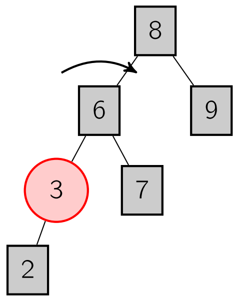


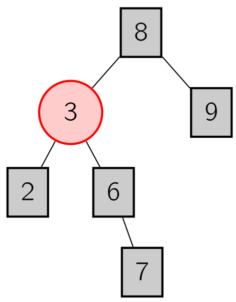
Lösche 6:

Wir löschen den Knoten mit dem nächst größeren Wert 7 und fügen diesen Wert dann in den zu löschenden Knoten ein.
Lösche 7:
Fall 4: umfärben

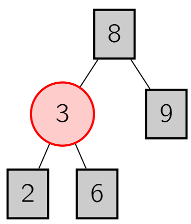


rotiere 6 nach rechts

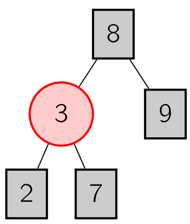




ersetze 7 durch rechtes Kind



füge 7 in den zu löschenden Knoten ein



Aufgabe 5 (Baumhöhe vollständiger binärer Bäume): (3 Punkte)

Wie stellen zunächst einige Begrifflichenkeiten klar: Sei B ein Baum mit Wurzel w, und sei v ein beliebiger Knoten von B. Die *Tiefe* von v in B ist definiert als der Abstand von v zu w. Die *Höhe* von B ist definiert als die Tiefe des tiefsten Knotens von B. Die *Ebene* k von B ist die Menge aller Knoten von B der Tiefe k. Sei h die Höhe von B, dann nennen wir B *vollständig*, genau dann wenn wenn B auf jeder Ebene k für $0 \leq k \leq h$ maximal viele Knoten enthält.
Beweisen Sie nun, dass die Höhe eines vollständigen binären Baumes mit n Knoten in $\Theta(\log(n))$ liegt.
Hinweis: Folgende Teilschritte des Beweises könnten sich als hilfreich erweisen:

- Beweisen Sie, wie viele Knoten ein vollständiger Baum auf jeder Ebene enthält.
- Beweisen Sie, wie viele Knoten ein vollständiger Baum enthält.

Lösung:

Wir beweisen zunächst mittels vollständiger Induktion, dass ein vollständiger binärer Baum auf der Ebene k genau 2^k Knoten enthält:

Induktionsanfang: Die Ebene 0 enthält alle Knoten mit Abstand 0 zur Wurzel. Damit enthält die Ebene 0 genau $1 = 2^0$ Knoten, nämlich nur die Wurzel selbst.

Induktionshypothese: Die Ebene k enthält 2^k Knoten.

Induktionsschritt: Laut Induktionshypothese enthält die Ebene k genau 2^k Knoten. Die Knoten der Ebene $k+1$ sind genau alle Kinder aller Knoten der Ebene k. Da jeder Knoten maximal 2 Kinder haben kann, enthält die Ebene $k+1$ somit genau $2 \cdot 2^k = 2^{k+1}$ Knoten. Die Aussage ist somit auch für $k+1$ und damit für alle $k \in \mathbb{N}$ gezeigt.

Als nächstes beweisen wir, dass ein vollständiger binärer Baum der Höhe h genau $2^{h+1} - 1$ viele Knoten enthält. Dazu müssen wir lediglich die Anzahl aller Knoten der Ebenen 0 bis h aufsummieren. Da wir zuvor bewiesen haben, dass die Ebene k genau 2^k Knoten enthält, ergibt sich die Summenformel

$$\sum_{k=0}^{h} 2^i$$

für die wir im Tutorium die geschlossene Form $2^{h+1} - 1$ bewiesen haben.

Sei n nun die Anzahl der Knoten eines vollständigen binären Baumes der Höhe h. Dann gilt

$$\begin{aligned} 2^{h+1} - 1 &= n \\ 2^{h+1} &= n + 1 \\ h + 1 &= \log_2(n+1) \\ h &= \log_2(n+1) - 1 \end{aligned}$$

Es bleibt nun nur noch zu zeigen, dass $h \in \Theta(\log(n))$ liegt. Wir bestimmen dazu den Grenzwert

$$\lim_{n\to\infty} \frac{\log_2(n+1)}{\log_a(n)}$$

für beliebiges $0 < a \neq 1$:

$$\begin{aligned} \lim_{n\to\infty} \frac{\log_2(n+1)}{\log_a(n)} &= \lim_{n\to\infty} \frac{\frac{\ln(n+1)}{\ln(2)}}{\frac{\ln(n)}{\ln(a)}} = \frac{\ln(a)}{\ln(2)} \cdot \lim_{n\to\infty} \frac{\ln(n+1)}{\ln(n)} \\ &\overset{\text{l'Hospital}}{=} \frac{\ln(a)}{\ln(2)} \cdot \lim_{n\to\infty} \frac{n}{n+1} = \frac{\ln(a)}{\ln(2)} = c \neq 0 \end{aligned}$$

Aufgabe 6 (Inorder-Traversierung binärer Suchbäume): (3 + 3 Punkte)

Es soll bewiesen werden, dass die Inorder-Traversierung alle Schlüssel eines binären Baumes B genau dann in sortierter Reihenfolge ausgibt, wenn B ein Suchbaum ist. Führen sie dazu die folgenden Schritte durch:

a) Beweisen Sie mittels struktureller Induktion über den Baumaufbau, dass die Inorder-Traversierung eines binären Suchbaumes alle Schlüssel des Baumes in sortierter Reihenfolge ausgibt!

b) Beweisen Sie, dass ein binärer Baum B ein Suchbaum ist, wenn seine Inorder-Traversierung die Schlüssel in sortierter Reihenfolge ausgibt!

Lösung:

a) Wir beweisen die Aussage mittels struktureller Induktion über den Baumaufbau.

Induktionsanfang: Als Basisfall verwenden wir einen binären Suchbaum, der nur einen einzigen Knoten mit Schlüssel k enthält. Die Inorder-Traversierung dieses Baumes ist die Sequenz k. Diese Sequenz der Länge 1 ist offensichtlich sortiert.

Induktionshypothese: Seien B_L und B_R zwei beliebige aber feste binäre Suchbäume. Dann gibt die Inorder-Traversierung des Baumes B_L die Schlüssel des Baumes B_L und die Inorder-Traversierung des Baumes B_R die Schlüssel des Baumes B_R in sortierter Reihenfolge aus.

Induktionsschritt: Sei nun k ein Schlüssel und seien B_L und B_R zwei beliebige aber feste binäre Suchbäume, sodass alle Schlüssel des Baumes B_L höchstens so groß wie k und alle Schlüssel des Baumes B_R mindestens so groß wie k sind. Dann ist der Baum B, dessen Wurzel den Schlüssel k hat und als linken Teilbaum den Baum B_L und als rechten Teilbaum den Baum B_R hat, wiederum ein binärer Suchbaum.

Die Inorder-Traversierung des Baumes B ist nun wie folgt: Zunächst wird die Inorder-Traversierung des linken Teilbaumes B_L ausgegeben. Diese ist nach Induktionshypothese sortiert. Anschließend wird der Schlüssel der Wurzel von B, d.h. k, ausgegeben. Laut Annahme ist k mindestens so groß wie alle Schlüssel in B_L; die Sortiertheit der Ausgabe bleibt also unverletzt. Als letztes wird wird die Inorder-Traversierung des rechten Teilbaumes B_R ausgegeben. Auch diese ist nach Induktionshypothese sortiert. Laut Annahme ist k höchstens so groß wie alle Schlüssel in B_R; die Sortiertheit der Ausgabe bleibt also auch in diesem letzten Schritt unverletzt.

Insgesamt wurden somit alle Schlüssel von B in sortierter Reihenfolge ausgegeben.

b) Wir beweisen diese Aussage mittels Induktion über die Länge der Inorder-Traversierung.

Induktionsanfang: Als Basisfälle verwenden wir die Längen 0 und 1. Eine Inorder-Traversierung der Länge 0 ist die des leeren Baumes und die Inorder-Traversierung der Länge 1 ist die eines Baumes mit nur einem einzigen Element. Diese beiden Bäume erfüllen trivialerweise die Suchbaumeigenschaft.

Induktionshypothese: Seien B_L und B_R zwei beliebige aber feste binäre Bäume deren Inorder-Traversierungen v und w jeweils in sortierter Reihenfolge sind. Dann sind B_L und B_R binäre Suchbäume.

Induktionsschritt: Für den Induktionsschritt betrachten wir nun die (längere) Inorder-Traversierung des folgenden Baumes B: B hat eine Wurzel mit Schlüssel k und linkem Teilbaum B_L und rechtem Teilbaum B_R. Bei der Inorder-Traversierung von B werden nun zunächst alle Schlüssel aus B_L – wir nennen diese Sequenz v –, dann k und anschließend alle Schlüssel aus B_R – wir nennen diese Sequenz w – ausgegeben. Die Inorder-Traversierung ist also gegeben durch vkw. Sei nun vkw in sortierter Reihenfolge. Dann sind alle Schlüssel, die in v enthalten sind, höchstens so groß wie k und damit auch alle Schlüssel in B_L höchstens so groß wie k. Analog dazu sind alle Schlüssel, die in w enthalten sind, mindestens so groß wie k und damit auch alle Schlüssel in B_R mindestens so groß wie k. Insgesamt erfüllt damit auch der Baum B die Suchbaumeigenschaft.

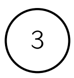
Aufgabe 7 (AVL Bäume): (7 Punkte)

Führen Sie folgenden Operationen beginnend mit einem anfangs leeren *AVL-Baum* aus und geben Sie die entstehenden Bäume nach jeder *Einfüge-* und *Löschoperation* sowie jeder *Rotation* an. Markieren Sie außerdem zu jeder Rotation, welcher Knoten in welche Richtung rotiert wird:

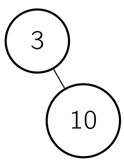
1. 3 einfügen
2. 10 einfügen
3. 6 einfügen
4. 1 einfügen
5. 2 einfügen
6. 4 einfügen

Lösung:

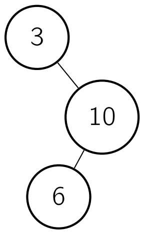
füge 3 ein



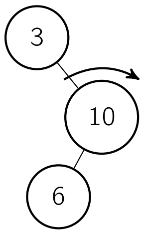
füge 10 ein

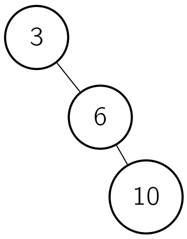


füge 6 ein

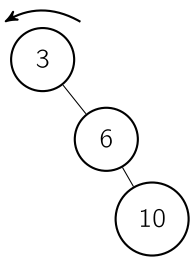


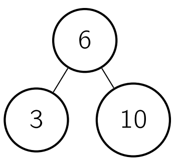
rotiere 10 nach rechts



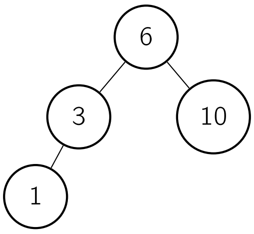


rotiere 3 nach links

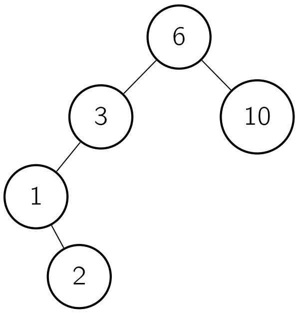




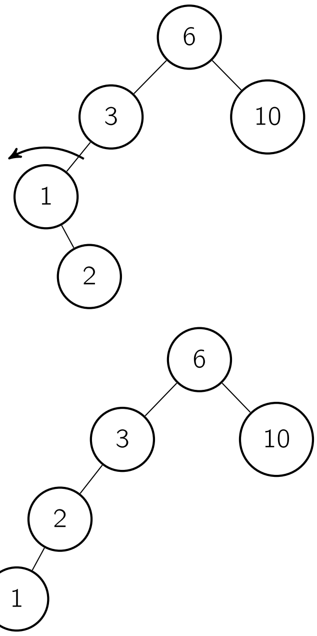
füge 1 ein



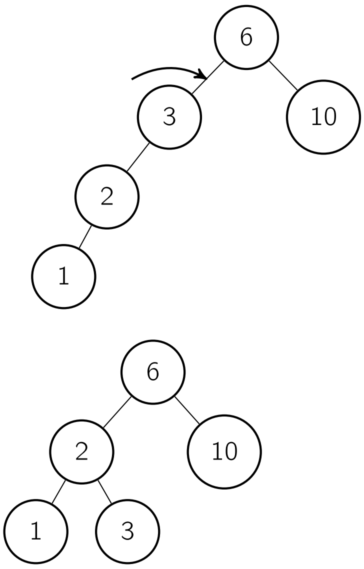
füge 2 ein



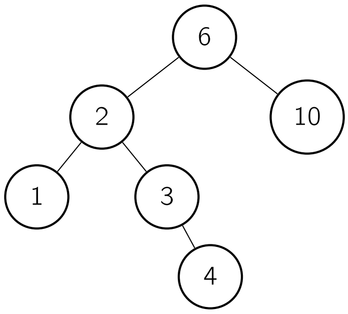
rotiere 1 nach links



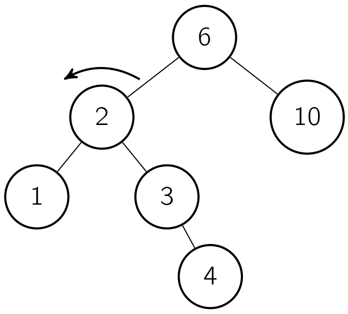
rotiere 3 nach rechts

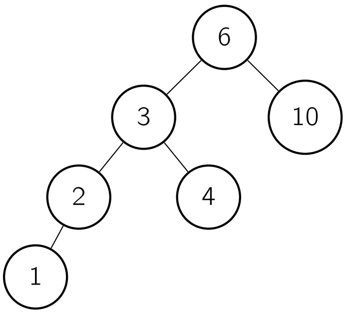


füge 4 ein

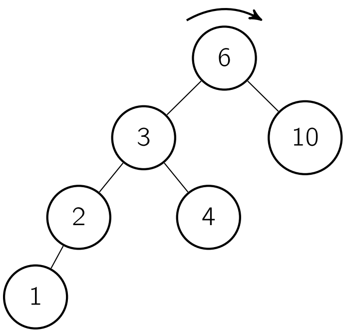


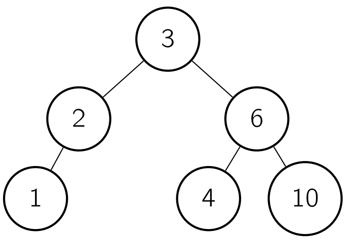
rotiere 2 nach links





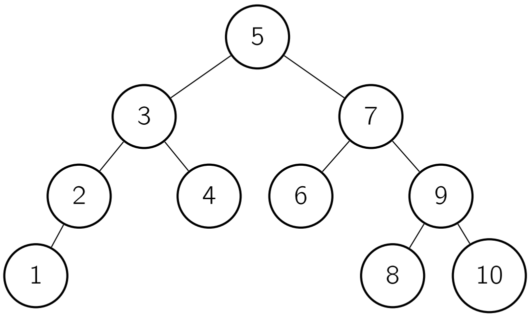
rotiere 6 nach rechts



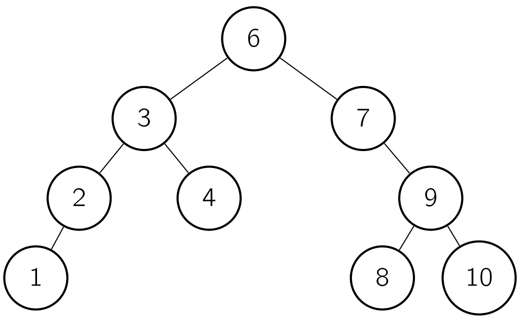


Aufgabe 8 (AVL Bäume): (2 Punkte)

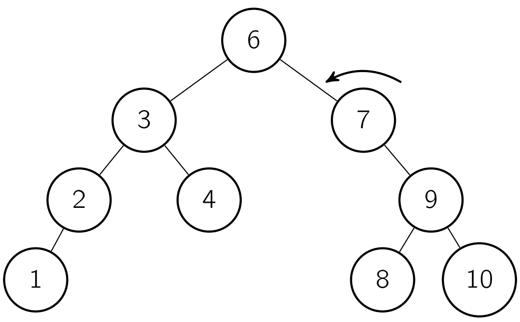
Löschen Sie den Wert 5 aus dem folgenden *AVL-Baum* und geben Sie die entstehenden Bäume nach jeder *Löschoperation* sowie jeder *Rotation* an. Markieren Sie außerdem zu jeder Rotation, welcher Knoten in welche Richtung rotiert wird:

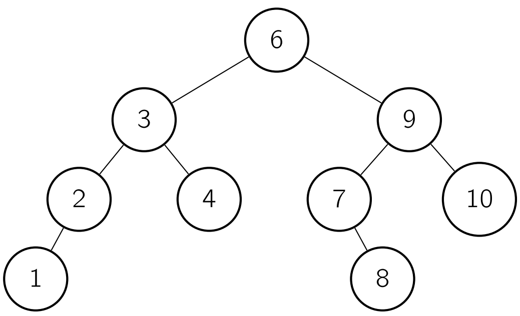


Lösung:
entferne 5



rotiere 7 nach links





Aufgabe 9 (Rot-Schwarz Bäume): (7 Punkte)

Führen Sie die folgenden Operationen beginnend mit einem anfangs leeren *Rot-Schwarz-Baum* aus und geben Sie die entstehenden Bäume nach

- jeder *Einfügeoperation*,
- jeder *Löschoperation*,
- jeder *Rotation* sowie
- jeder *Umfärbung* an.

Markieren Sie außerdem zu jeder Rotation, welcher Knoten in welche Richtung rotiert wird. Mehrere Umfärbungen können Sie in einem Schritt zusammenfassen. Beachten Sie, dass rote Knoten rund und schwarze Knoten eckig dargestellt werden.

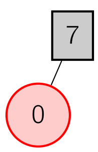
1. 7 einfügen
2. 0 einfügen
3. 4 einfügen
4. 9 einfügen
5. 8 einfügen
6. 6 einfügen

Lösung:

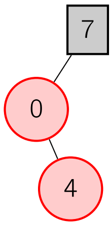
füge 7 ein

7

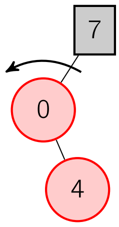
füge 0 ein

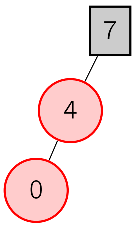


füge 4 ein

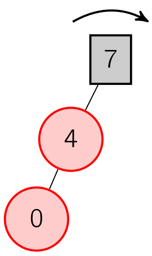


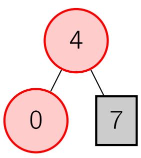
rotiere 0 nach links



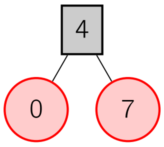


rotiere 7 nach rechts

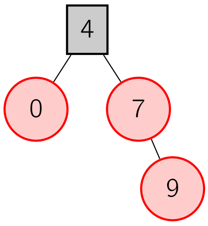




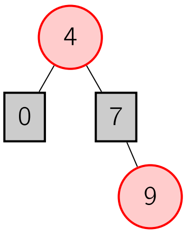
Fall 3: umfärben



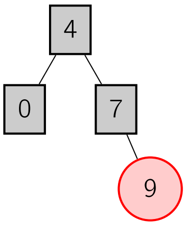
füge 9 ein



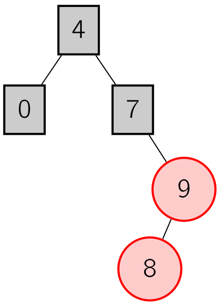
Fall 1: umfärben



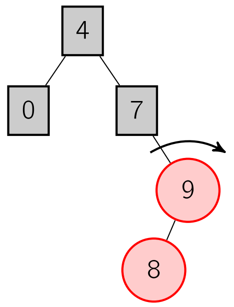
Wurzel schwarz färben

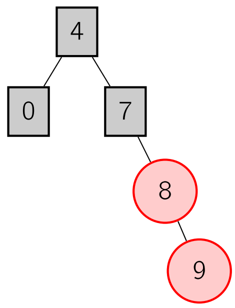


füge 8 ein

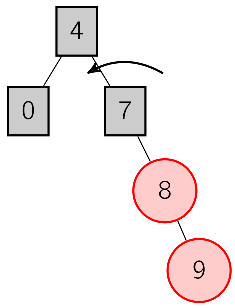


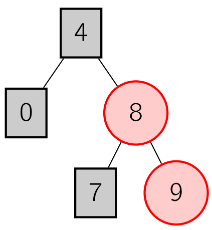
rotiere 9 nach rechts



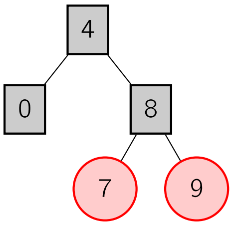


rotiere 7 nach links

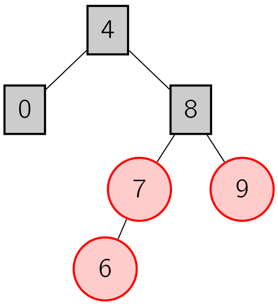




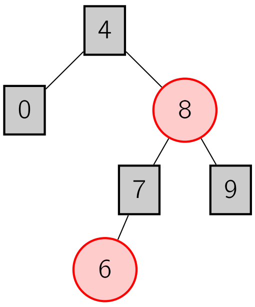
Fall 3: umfärben



füge 6 ein

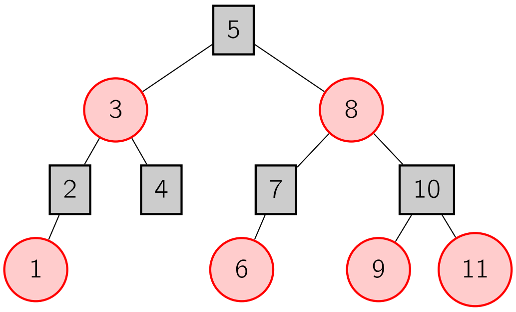


Fall 1: umfärben



Aufgabe 10 (Rot-Schwarz Bäume): (3 Punkte)

Betrachten Sie den folgenden *Rot-Schwarz-Baum*:



Führen Sie beginnend mit diesem Rot-Schwarz-Baum die folgenden Operationen aus und geben Sie die entstehenden Bäume nach

- jeder *Einfügeoperation*,
- jeder *Löschoperation*,
- jeder *Rotation* sowie
- jeder *Umfärbung* an.

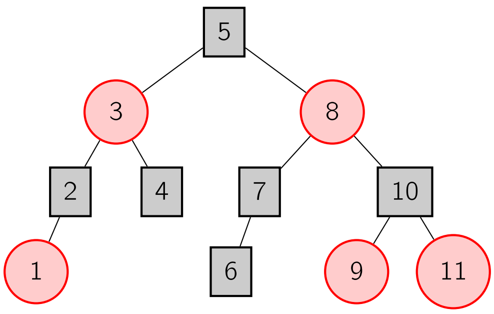
Markieren Sie außerdem zu jeder Rotation, welcher Knoten in welche Richtung rotiert wird. Mehrere Umfärbungen können Sie in einem Schritt zusammenfassen. Beachten Sie, dass rote Knoten rund und schwarze Knoten eckig dargestellt werden.

1. 7 löschen

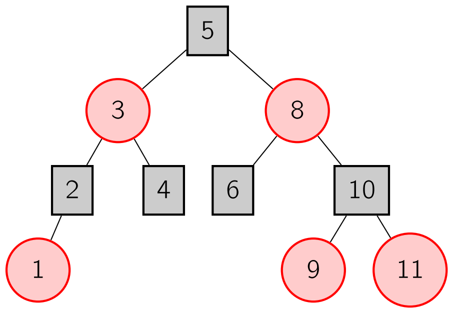
2. 6 löschen

Lösung:

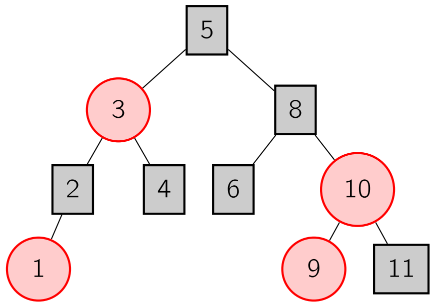
Lösche 7:
Knoten schwarz färben



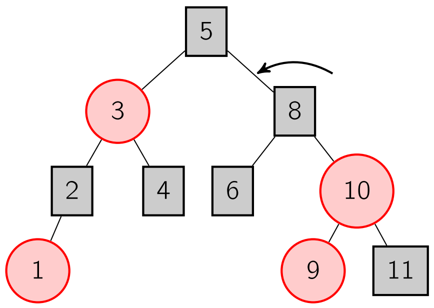
ersetze 7 durch linkes Kind

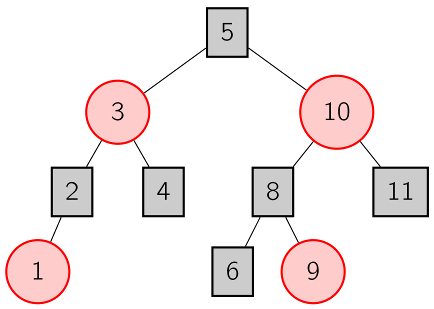


Lösche 6:
Fall 4: umfärben



rotiere 8 nach links





ersetze 6 durch rechtes Kind