

# Datenstrukturen und Algorithmen

Abgabe: 31.05.2017

Georg C. Dorndorf Matr.Nr. 366511  
Adrian C. Hinrichs Matr.Nr. 367129

# 6	# 7	# 8	# 9	# 10	# 11	# 12	Σ

## Aufgabe 6.b

## Aufgabe 6

### Aufgabe 6.a

Sei  $B$  ein Baum gegeben durch 1.

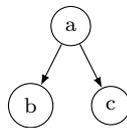


Abbildung 1: Suchbaum  $B$

Für jeden Baum, der Form  $B$  gilt falls es ein Suchbaum ist:

$$b \leq a < c^1 \tag{I}$$

Die Inorder-Traversierung gibt nun die Knoten in der Reihenfolge **linker Teilbaum, Wurzel, rechter Teilbaum** aus beziehungsweise an den entsprechenden Stellen die Blätter. Es ist dabei unerheblich ob der Baum ausgeglichen ist. Wir beweisen nun, dass die Inorder-Traversierung jeden Suchbaum  $S$  in sortierter Reihenfolge ausgibt.

(IA) :  $Höhe(S) = 0$ : Trivialerweise ist ein einzelnes Element stets sortiert.  $Höhe(S) = 1$ : Da für jeden Suchbaum  $I$  gilt werden die Elemente offensichtlich sortiert ausgegeben. ✓

(IV) : Dies gelte nun für einen beliebig aber festen Suchbaum mit Höhe  $h \in \mathbb{N}$

$$(IS) : Höhe(S) \mapsto h + 1$$

Dieser Baum  $S$  lässt sich in seine Wurzel, die trivialerweise sortiert ist und zwei oder (falls die Wurzel nur ein Kind hat in einen) Unterbaum  $S_{links}, S_{rechts}$  mit der Höhe  $h$  aufteilen. Für diese unterbäume gilt dann, dass ihre Höhe  $h$  ist. Sie werden also nach (IV) korrekt sortiert ausgegeben. Wobei für alle Elemente von  $S_{links}$ , die mithilfe der Inorder-Traversierung ausgegeben werden gilt, dass sie kleiner als die Wurzel von  $S$  sind (I). Für alle Elemente von  $S_{rechts}$  gilt, dass sie größer als die Wurzel von  $S$  sind. Nach der Definition der Inorder-Traversierung wird der Baum  $S$  also korrekt sortiert ausgegeben. Wir haben also mithilfe der Strukturellen Induktion nach  $h \in \mathbb{N}$  gezeigt, dass jeder Suchbaum  $S$  mithilfe der Inorder Traversierung sortiert ausgegeben wird. □

Sei die Ausgabe einer Inorder-Traversierung eines Baums  $B$  durch das  $n$ -Tupel  $w$  gegeben und sortiert. Für  $(w_1, \dots, w_n)$  gilt demnach  $(w_1 \leq \dots \leq w_n)$ .

(IA) : Länge( $w$ ) = 1: Eine einzelne Wurzel erfüllt nach Def. die Eigenschaften eines binären Suchbaums. ✓

(IV) : Dies gelte nun für eine beliebig aber feste Ausgabe  $w$  einer Inorder-Traversierung eines Baumes  $B$ .

(IS) : Länge( $w$ )  $\mapsto n + 1$ : Dann lässt sich  $w$  in  $grp$  unterteilen wobei  $r$  die Wurzel eines Baumes  $B$  repräsentiert also gilt  $\exists k \in \mathbb{N}_{\leq n+1}$  mit  $w_k = r$ . Es sind dann  $q = (w_0, \dots, w_{k-1})$  und  $p = (w_{k+1}, \dots, w_{n+1})$ . Es folgt, dass Länge( $q$ )  $\leq n$ , Länge( $p$ )  $\leq n$ . Seien  $B_{links}$  und  $B_{rechts}$  die von  $q, p$  erzeugten Bäume. Nach (IV) gilt, dass die zu  $p, q$  zugehörigen Teilbäume  $B_{links}, B_{rechts}$  sortiert sind. Nach der Definition eines Suchbaums ist dann auch der durch die Konkatenation  $grp$  erzeugte Baum ein binärer Suchbaum. □

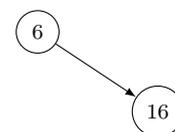
## Aufgabe 7

1)



Abbildung 2: Baum nach der Operation 6 einfügen

2)



□ Abbildung 3: Baum nach der Operation 16 einfügen

3)

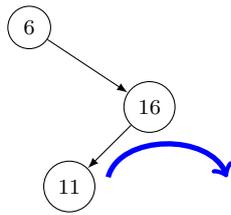


Abbildung 4: Baum nach der Operation 11 einfügen. Der blaue Pfeil markiert die auszuführende Rotationsoperation

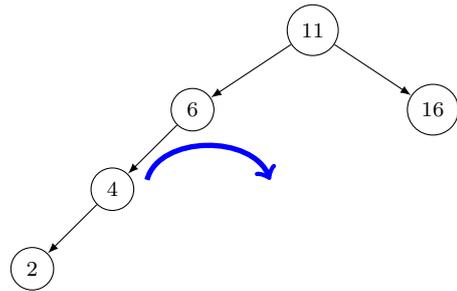


Abbildung 9: Baum nach der Operation 4 einfügen. Der blaue Pfeil markiert die auszuführende Rotationsoperation

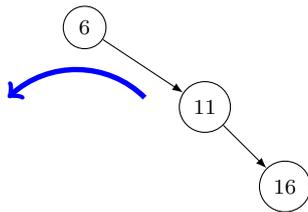


Abbildung 5: Baum nach der ersten Rotation. Der blaue Pfeil markiert die auszuführende Rotationsoperation

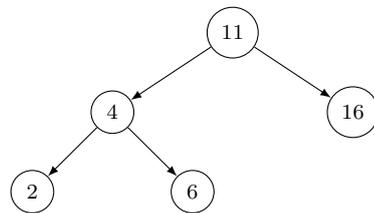


Abbildung 10: Baum nach der Operation 4 einfügen.

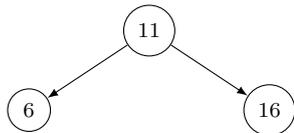


Abbildung 6: Baum nach der Rebalancierung

6)

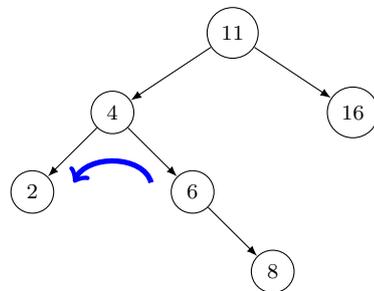


Abbildung 11: Baum nach der Operation 8 einfügen. Der blaue Pfeil markiert die auszuführende Rotationsoperation

4)

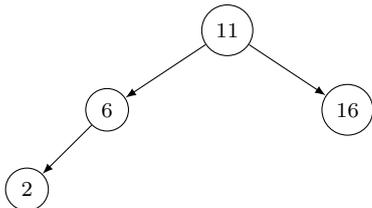


Abbildung 7: Baum nach der Operation 2 einfügen.

5)

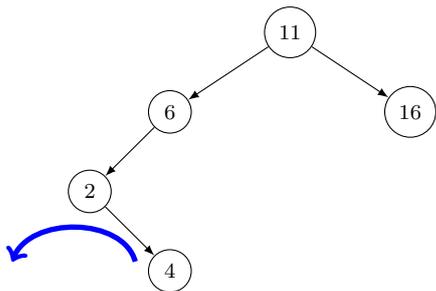


Abbildung 8: Baum nach der Operation 4 einfügen. Der blaue Pfeil markiert die auszuführende Rotationsoperation

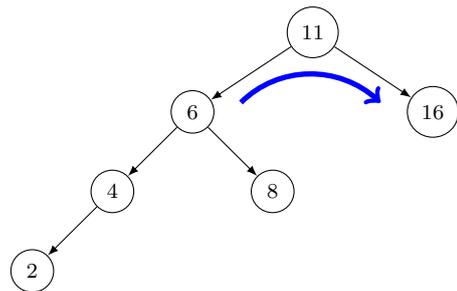


Abbildung 12: Baum nach der Rotate-Operation. Der blaue Pfeil markiert die auszuführende Rotationsoperation

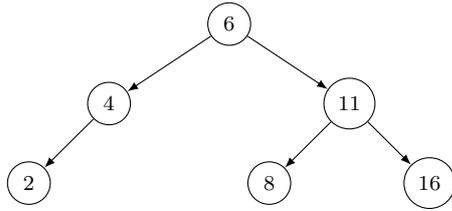


Abbildung 13: Baum nach der Rotate-Operation

## Aufgabe 8

### Aufgabe 8.a

Eine Reihenfolge in der die Zahlen zum AVL-Baum hinzugefügt werden können, damit mindestens eine Rotation nötig ist und der geforderte Baum entsteht ist: 10, 12, 8, 4, 2, 6

### Aufgabe 8.b

Wenn man in den AVL-Baum aus a) einen Knoten mit dem Schlüssel 11 einfügt werden die Anforderungen der Aufgabenstellung erfüllt, da  $11 \leq 12 \wedge 11 \in \mathbb{N}$ .

## Aufgabe 9

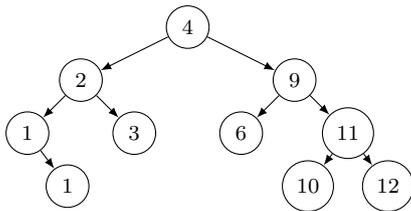


Abbildung 14: Baum in der Ausgangslage.

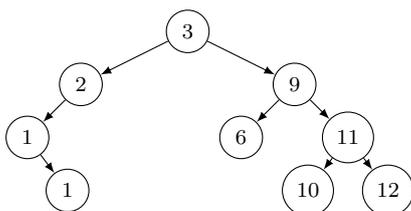


Abbildung 15: Baum nach der Löschoption für den Wert 4.

Der Knoten mit dem Wert 4 wurde gelöscht und durch den größten Knoten des linken Teilbaumes (in diesem Fall 3) ersetzt.

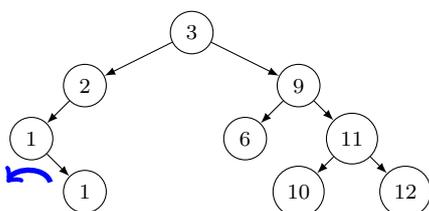


Abbildung 16: Baum nach der Löschoption und vor der ersten Rotation.

Der Pfeil beschreibt die Rotation, die der Algorithmus zur Balancierung des Baumes als nächstes vornimmt. Hier wird nun der tiefste Knoten, der die Balanciertheit nicht erfüllt nach links oben rotiert.

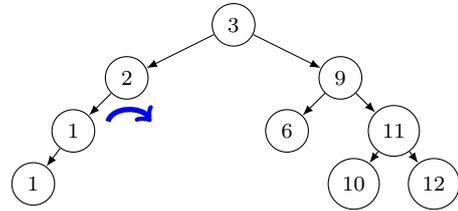


Abbildung 17: Baum nach der ersten und vor der zweiten Rotation.

Nun wird der ander Knoten mit dem Wert 1 nach rechts oben rotiert. Danach ist der AVL-Baum balanciert.

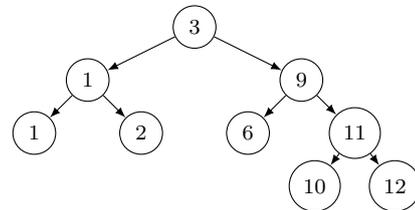


Abbildung 18: Baum am Ende der gesamten Löschoption.

## Aufgabe 10

Sei  $\phi := \frac{1+\sqrt{5}}{2}$ . Sei  $T$  ein AVL-Baum mit  $n$  Knoten und der Höhe  $h$ . Sei  $N(h)$  eine Rekursionsformel für die minimale Anzahl an Knoten für einen Baum der Höhe  $h$ . Sei  $N(h)$  gegeben durch:

$$N(h) = \begin{cases} 1 & ,h = 0 \\ 2 & ,h = 1 \text{ (II)} \\ N(h-1) + N(h-2) + 1 & ,h \geq 2 \end{cases}$$

Der erste Fall ( $h = 0$ ) der rekursiven Funktion ergibt sich daraus, dass ein AVL-Baum der Höhe 0 aus einem Knoten und zwar der Wurzel besteht. Der Zweite Fall ( $h = 1$ ) folgt daraus, dass ein AVL-Baum der Höhe 1 aus mindestens einer Wurzel und einem Kind besteht also mindestens 2 Knoten hat. Der dritte Fall für  $h \geq 2$  folgt aufgrund der rekursiven Definition eines AVL-Baumes.

**Zu zeigen:**  $h \leq \log_{\phi}(n)$

**Beweis:** Wir formen diese Aussage um und beweisen dann eine Äquivalente mithilfe von vollständiger Induktion.

$$h \leq \log_{\phi}(n) \tag{III}$$

$$\Leftrightarrow h \leq \log_{\phi}(N(h)) \tag{IV}$$

$$\Leftrightarrow N(h) \geq \phi^h \tag{V}$$

Wir beweisen nun V mittels vollständiger Induktion 3) nach  $h \in \mathbb{N}_0$ .

$$(IA) : h = 0 : N(h) = 1 > \phi^0 - 1 = 0 \quad \checkmark \quad (VI)$$

$$h = 1 : N(h) = 2 > \phi^1 - 1 \quad \checkmark \quad (VII)$$

(IV) : Die Behauptung gelte nun für ein beliebig aber festes  $h \in \mathbb{N}_0$ . (VIII)

(IS) :  $h \mapsto h + 1$  : (IX)

$$N(h + 1) \stackrel{Def.}{=} 1 + N(h) + N(h - 1) \quad (X)$$

$$\stackrel{VIII}{\geq} 1 + \phi^h + \phi^{h-1} \quad (XI)$$

$$= \phi^h + \phi^{h+1} \quad (XII)$$

$$= \phi^{h-1} \cdot (\phi + 1) \quad (XIII)$$

$$= \phi^{h-1} \cdot \phi^2 \quad (XIV)$$

$$= \phi^{h+1} \quad (XV)$$

Somit haben wir mithilfe von vollständiger Induktion nach  $h \in \mathbb{N}_0$  die Aussage V bewiesen.

□

Mithilfe von III und IV ist die Aussage  $h \leq \log_\phi(n)$  bewiesen.

QED

### Aufgabe 11

Die Abbildungen geben den Zustand des Rot-Schwarz-Baumes nach der in der Beschreibung angegebene Operation an.

Die blauen Pfeile markieren jeweils die auszuführenden Rotations-Operationen.

1)

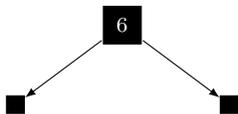


Abbildung 19: Einfüge-Operation von 6

2)

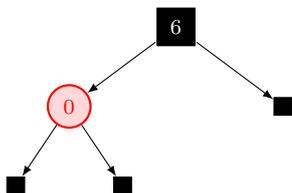


Abbildung 20: Einfüge-Operation von 0

4)

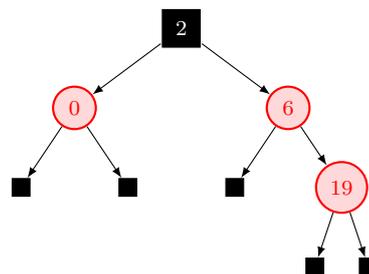


Abbildung 25: Einfüge-Operation von 19

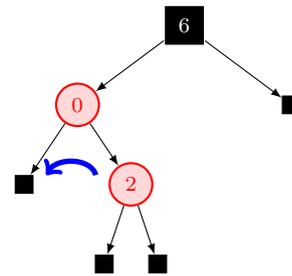


Abbildung 21: Einfüge-Operation von 2

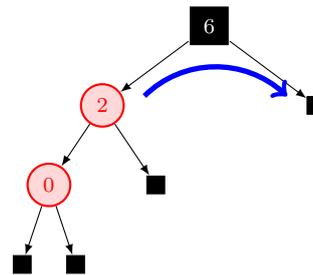


Abbildung 22: Left-Rotate über 0

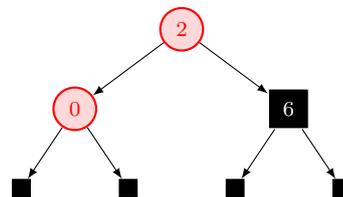


Abbildung 23: Right-Rotate über 6

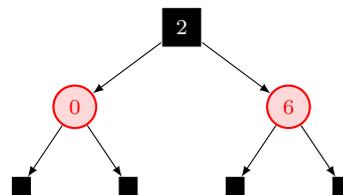


Abbildung 24: Umfärbung

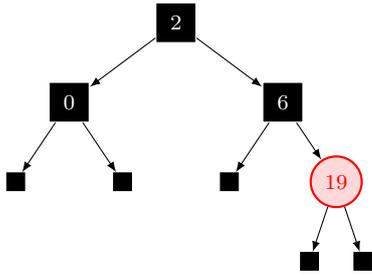


Abbildung 26: Umfärbung

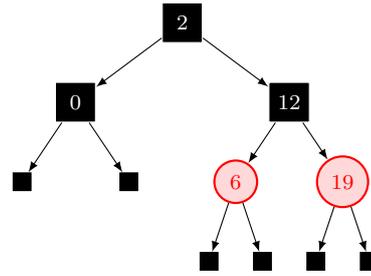


Abbildung 30: Umfärbung

5)

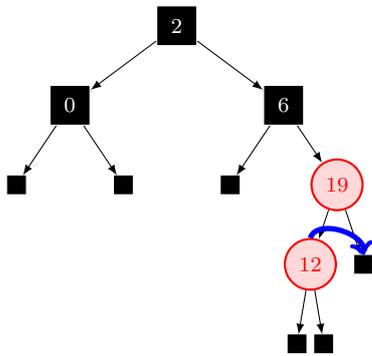


Abbildung 27: Einfüge-Operation von 12

6)

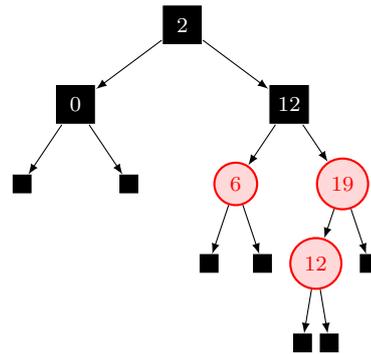


Abbildung 31: Einfüge-Operation von 12

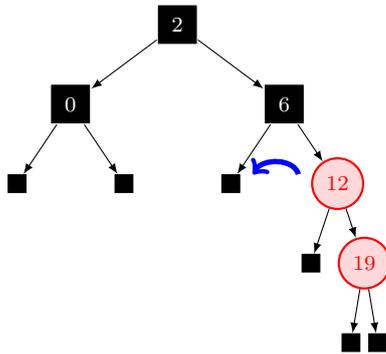


Abbildung 28: Right-Rotate über 19

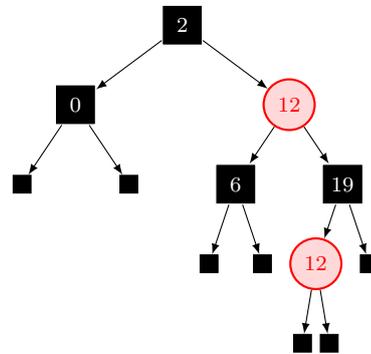


Abbildung 32: Umfärbung

### Aufgabe 12

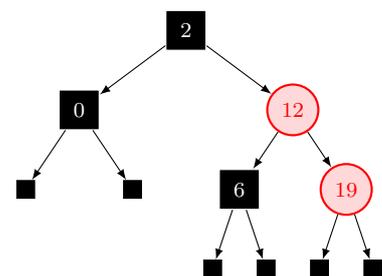


Abbildung 29: Left-Rotate über 6

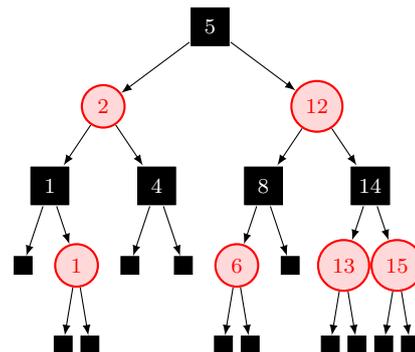


Abbildung 33: Baum zu Beginn

### Aufgabe 12.a

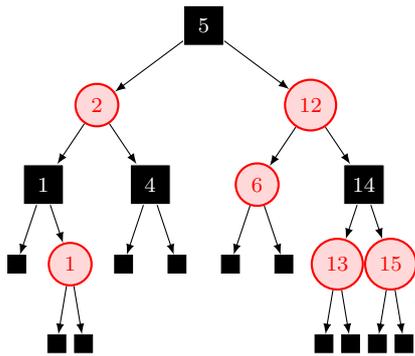


Abbildung 34: Nach Löschung der 8

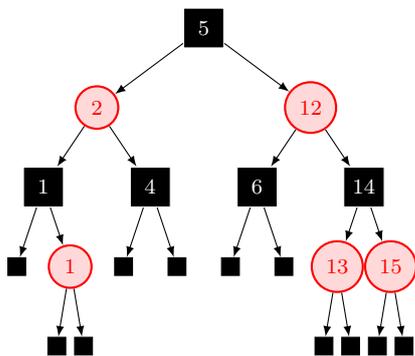


Abbildung 35: Nach Umfärbung

### Aufgabe 12.b

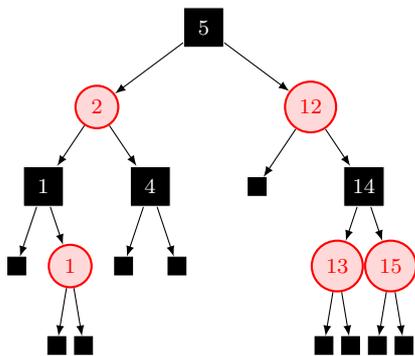


Abbildung 36: Nach Löschung der 6