Algorithmen und Datenstrukturen - Hausübung 05

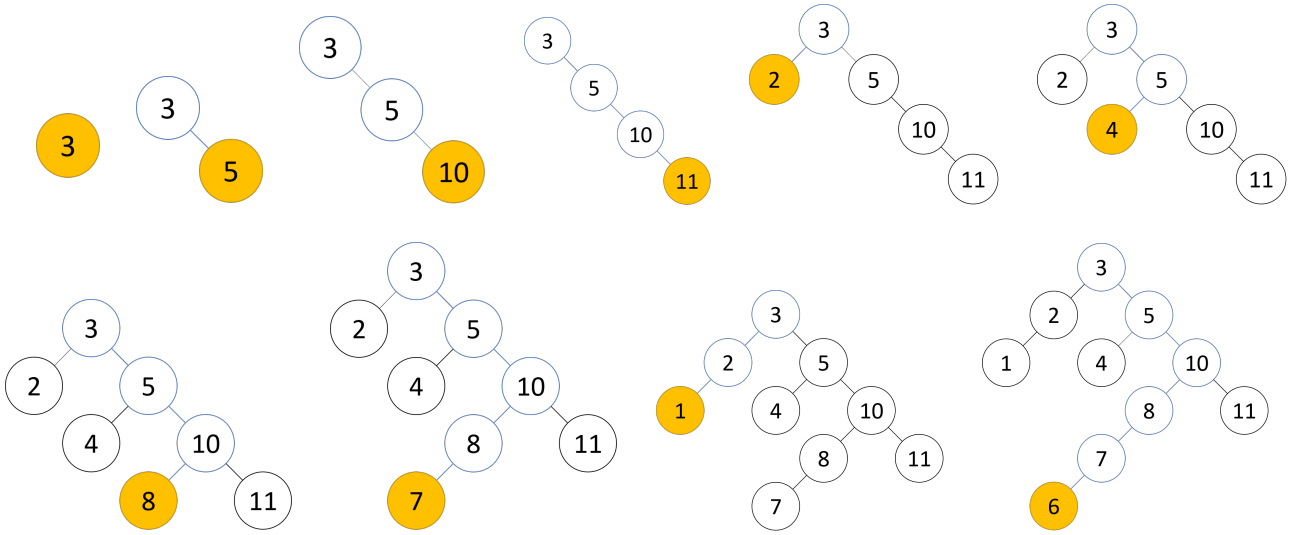
Gruppenmitglieder

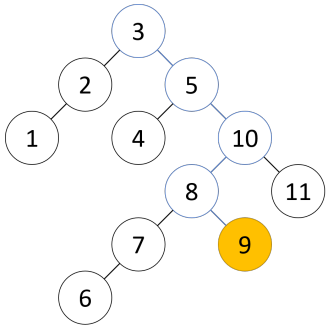
- Emre Berber (2957148)
- Christoph Berst (2743394)
- Jan Braun (2768531)

Inhaltsverzeichnis

H1

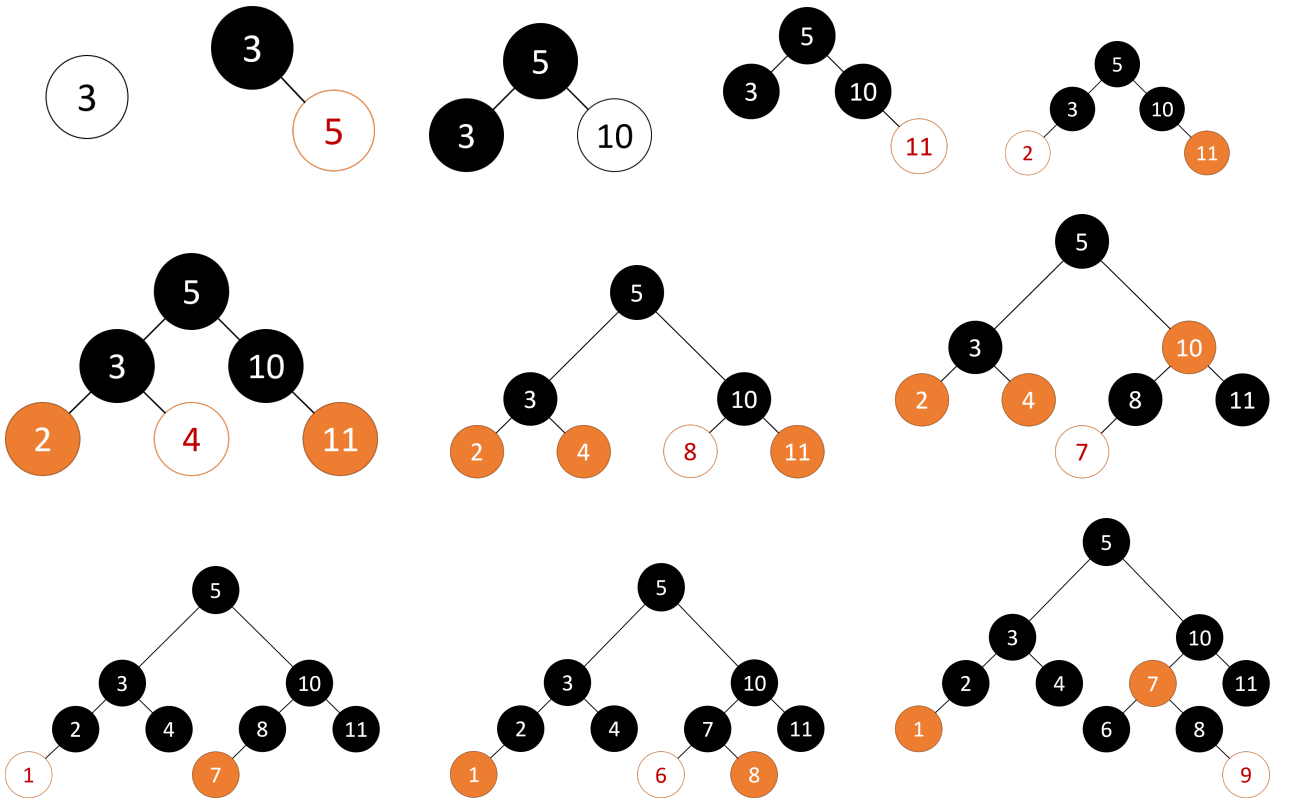
a)



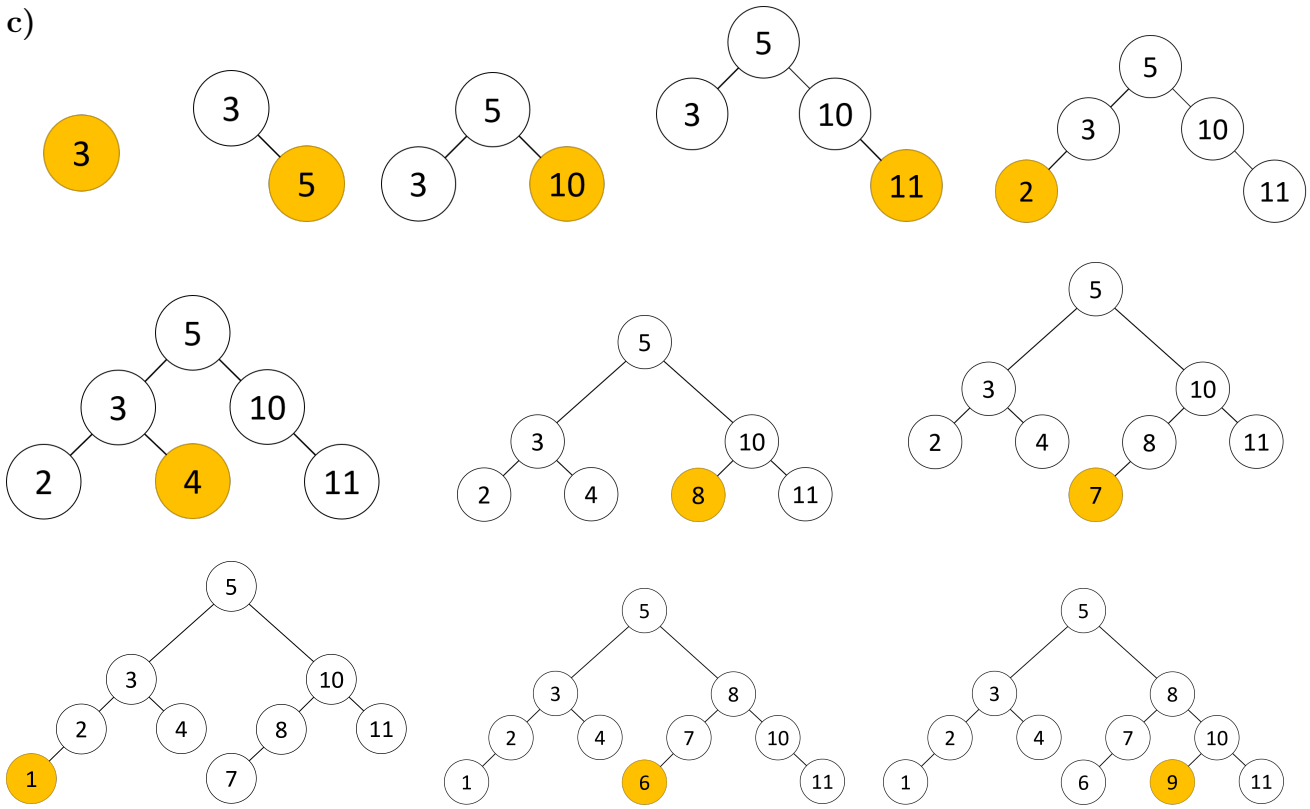


Wenn dies ein Rot-Schwarz-Baum werden soll, müssten die "1" und die "6" rot sein, da die "2" und die "7" über ihnen ansonsten nicht die selbe Anzahl an schwarzen Knoten in ihren Teilbäumen hätten. Somit wären die "2" und "7" darüber jedoch schwarz damit die "Nicht-Rot-Rot"-Regel erfüllt bleibt. Nun stellen wir fest, dass egal wie man man "5", "10" und "8" färben möchte einer von ihnen muss noch schwarz werden. Das würde jedoch bedeuten, dass der Teilbaum der "5" mindestens 2 schwarze Knoten auf seinen Pfaden hat. Der Teilbaum der "2" jedoch nur einen schwarzen Knoten hat. Daraus folgt, dass dies kein Rot-Schwarz-Baum sein kann.

b)



c)



Möglichkeiten diesen AVL-Baum als Rot-Schwarz-Baum darzustellen:

- Die Blätter "1", "6", "9" und "11" rot und den Rest schwarz.
- Die Blätter "1", "6", "9" und "11" rot. Die Knoten "3" und "8" rot. Der Rest ist schwarz.
- Die Knoten "1", "6" und "10" rot und den Rest schwarz.

H2

a)

Wenn wir um den Knoten z nach rechts rotieren, müssen wir im Wesentlichen nur 3 Knoten betrachten:

- x als linkes Kind von z ($x \leq z$)
- y das rechte Kind von x ($x \leq y$)
- und nartürlich z selbst

Da y im Teilbaum von x ist, welcher ein Unterbaum von z ist, ist auch $y \leq z$. Also ist $x \leq y \leq z$. Dies ist wichtig um zu erkennen, dass die Struktur des binären Suchbaum sich nicht ändert.
Im Algorithmus taucht auch noch p als Elter von z auf. Dies wird lediglich dafür gebraucht, um den Unterbaum nicht vom restlichen Baum zu trennen. Die wirkliche Veränderungen in der Binärbaum-Struktur geschehen nur in dem Unterbaum von z oder später von x.
Wir hängen nun y (ursprünglich rechter Unterbaum von x) statt x als linken Unterbaum an z und z wiederum als rechten Unterbaum an x, da y ja nun an z hängt.
Analysieren wir nun den Baum wie vor der Rotation stellen wir fest, dass z ein rechter Unterbaum von x und somit $x \leq z$ immer noch erfüllt ist. y ist ein linker Unterbaum von z und somit ist immer noch $y \leq z$. Und zu guter Letzt ist auch immer noch $x \leq y$, da y ein Unterbaum von z ist, welcher nun der rechte Unterbaum ($x \leq z$) von x ist.

b)

Betrachten wir die selben Knoten wie in der Aufgabe a).
Vor der Rotation befinden sich x und y im linken Unterbaum von z. Nun ersetzt x z als die Wurzel unseres Teilbaums und z und y wandern in den rechten Unterbaum. Die Wurzel wird also lediglich um einen anderen Knoten ersetzt. Nach der Implementation in der Vorlesung macht eine Rechtsrotation nur Sinn, wenn der Knoten z ein linkes Kind wie x besitzt. Daher wandert mindestens ein Knoten vom linken in den rechten Teilbaum. Zusätzlich kann y als Kind von x, falls es überhaupt existiert, in den rechten Teilbaum als Kind neues von z wandern. Also schrumpft der linke Teilbaum um mindestens einen Knoten und der rechte wächst um mindestens einen Knoten.
Bei der Rechtsrotation um z ist $L(x)_{danach} := L(z)_{vorher} - 1 - L(y)_{vorher}$, $L(y)_{danach} := L(y)_{vorher}$ und $L(z)_{danach} := L(z)_{vorher} - L(x)_{vorher} + L(y)_{vorher}$.
Bei jeder Rechtsrotation entfernen wir links von der Wurzel unseres Teilbäumen einen Unterbaum, jedoch fügen wir auf der rechten Seite wiederum einen Teilbaum y, falls dies überhaupt existierte, als linken Unterbäum wieder ein.

c)

qwertz

d)

Angenommen man hat eine Kette, die nur aus rechten Unterbäumen besteht, kann man nicht ohne Linksrotation dies in jeden anders strukturierten Binärbaum überführen. Dies gilt auch für ähnliche Bäume bei, denen mindestens ein Knoten keinen linken Unterbaum um diesen nach rechts zu rotieren. Daher kommen wir zu dem Schluss, dass dies nur mit Rechtsrotationen nicht möglich ist.

H3

a)

qwertz

b)

qwertz

c)

qwertz