

Algorithmen und Datenstrukturen

Sommersemester 2022

Woche 9

Kevin Angele, Tobias Dick, Oskar Neuhuber,
Andrea Portscher, Monika Steidl, Laurin Wischounig

Abgabe bis 31.05.2022 23:59

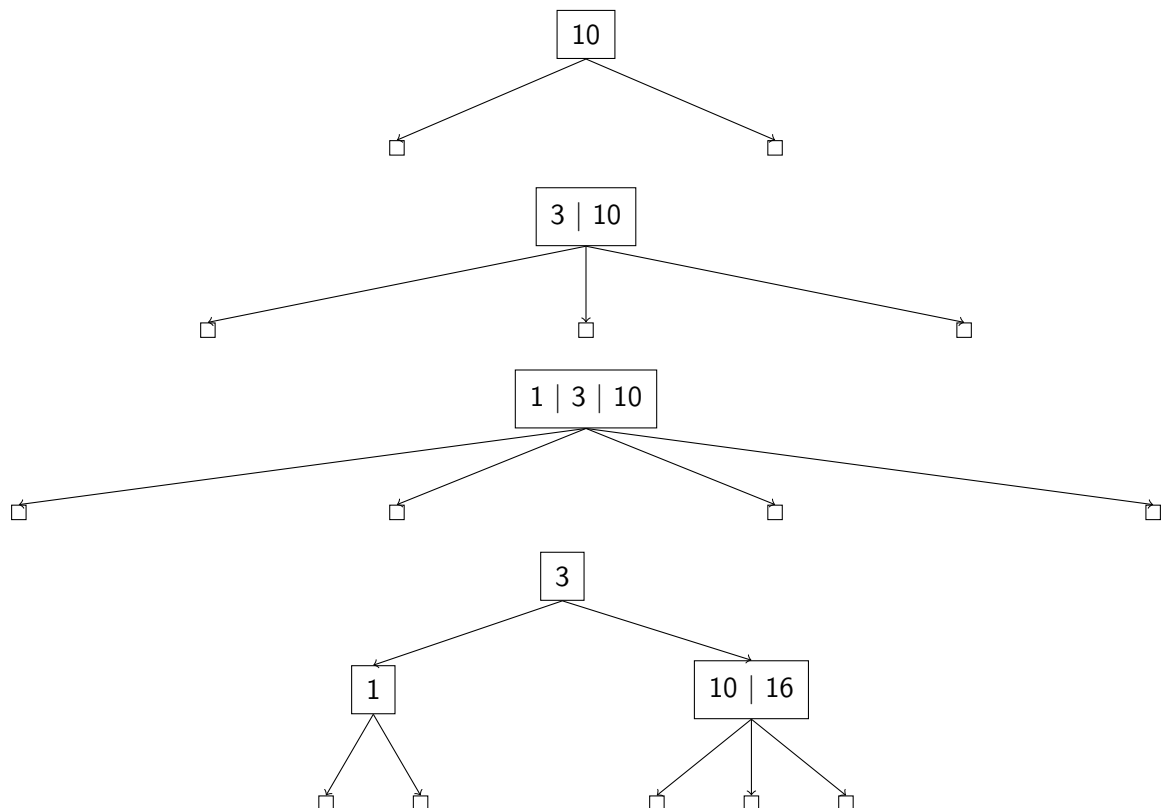
Besprechung im PS am 02.06.2022

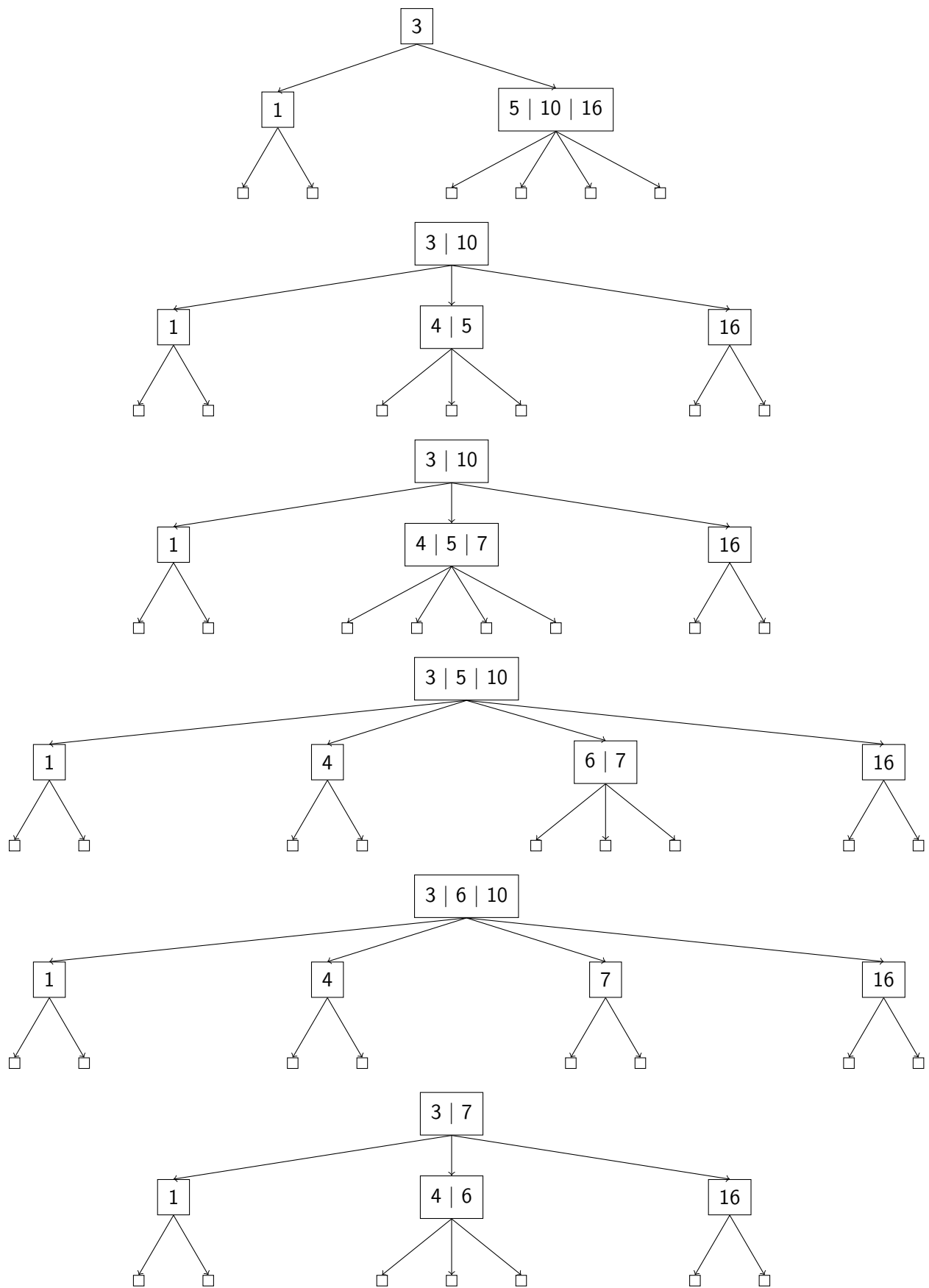
Aufgabe 1 (3 Punkte): 2,4-Baum

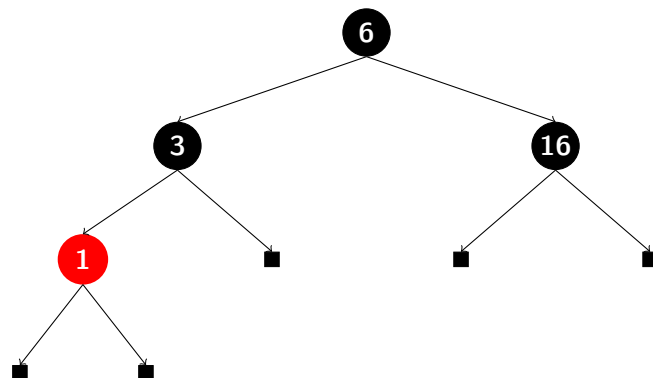
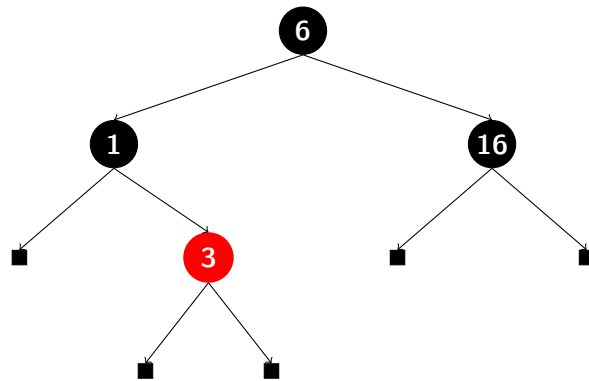
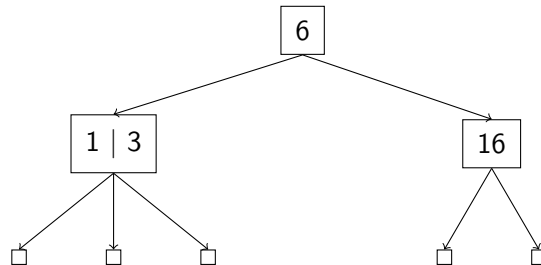
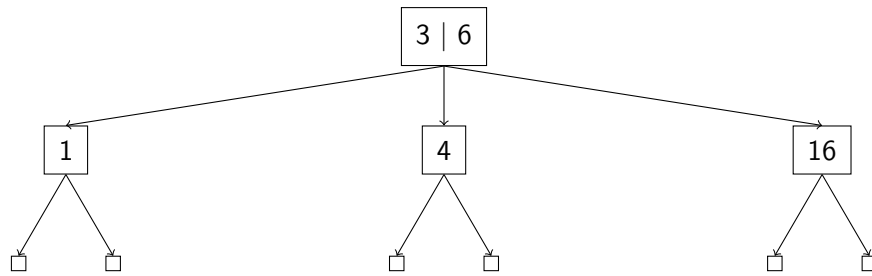
Starten Sie mit einem leeren 2,4-Baum. Führen Sie darauf die beschriebenen Operationen durch und zeigen sie den Baum nach jedem Schritt.

- Fügen Sie die Schlüssel $[10, 3, 1, 16, 5, 4, 7, 6]$ nacheinander ein.
- Löschen Sie die Knoten mit den Schlüssel $[5, 10, 7, 4]$.
- Zeichnen Sie den korrespondierenden Rot-Schwarz-Baum.

Lösung:





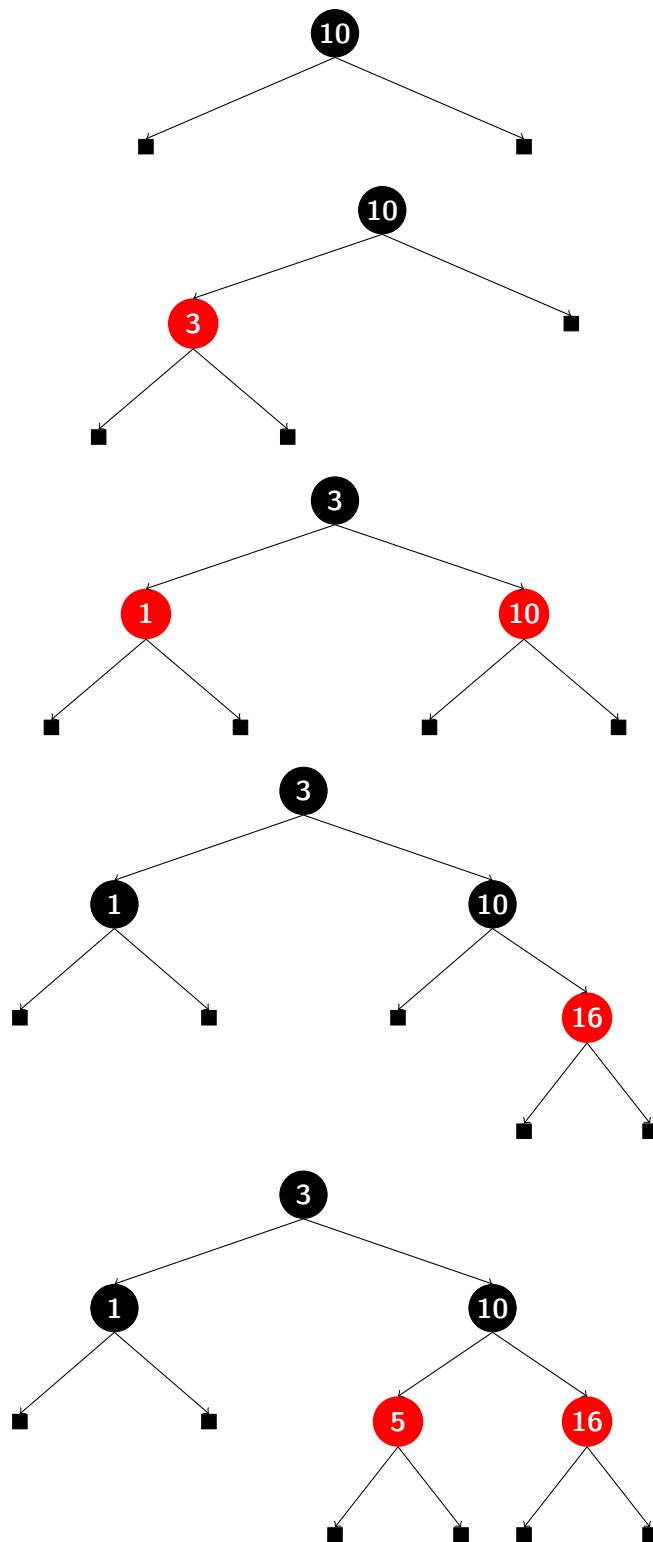


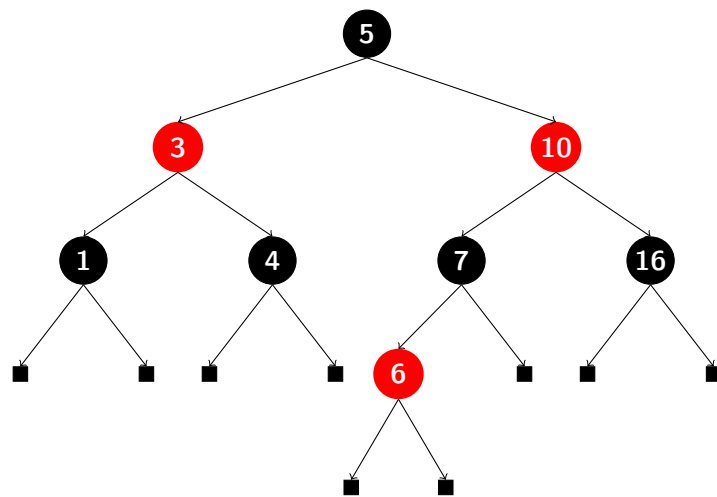
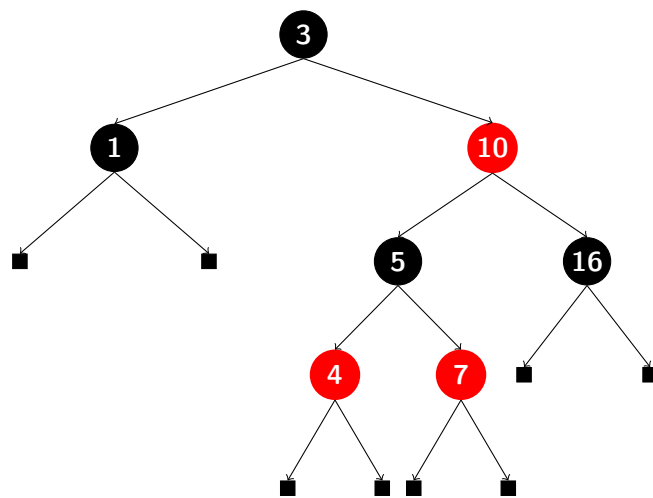
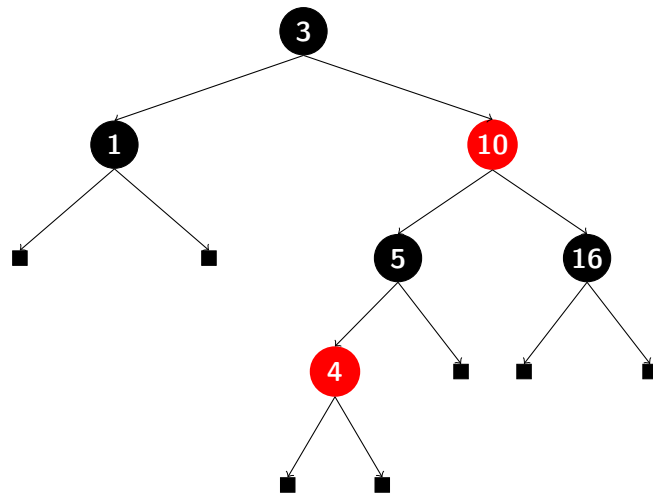
Aufgabe 2 (3 Punkte): Rot-Schwarz-Baum

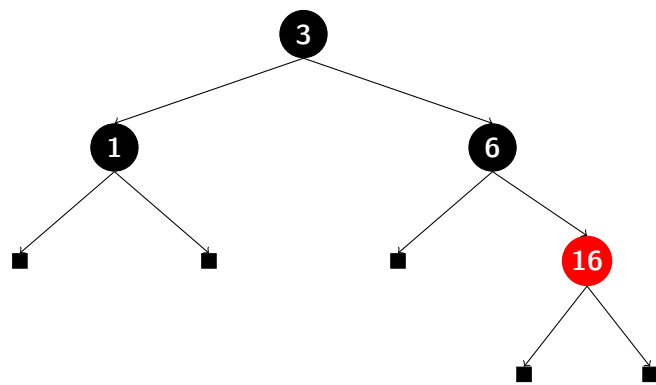
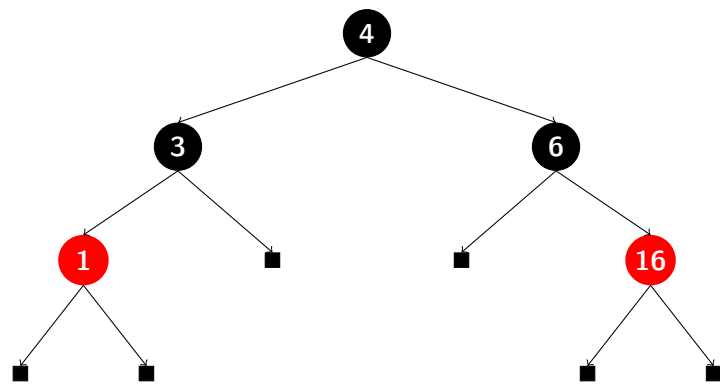
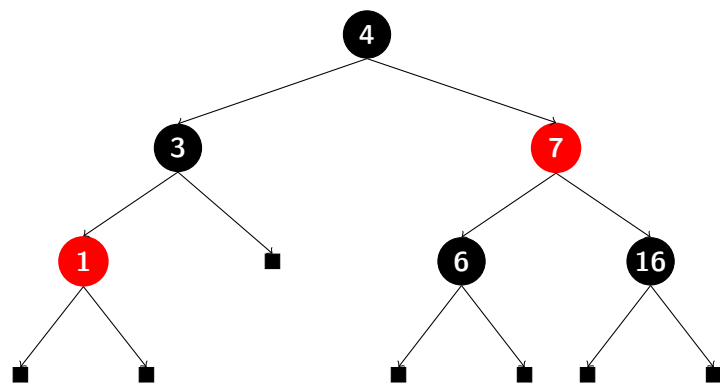
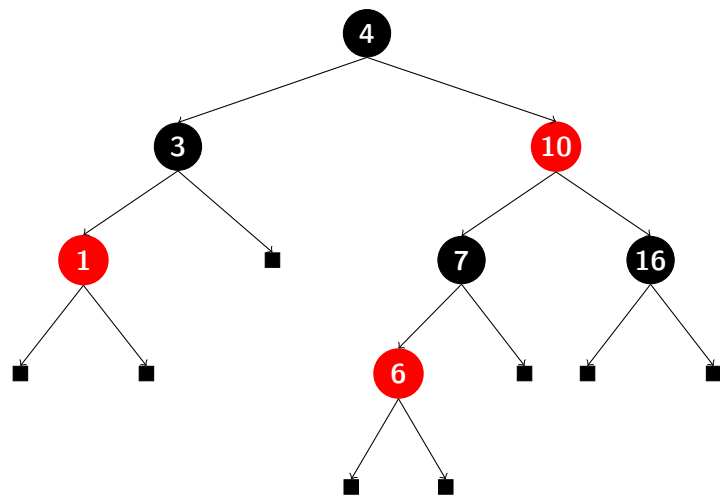
Starten Sie mit einem leeren Rot-Schwarz-Baum. Führen Sie darauf die beschriebenen Operationen durch und zeigen sie den Baum nach jedem Schritt.

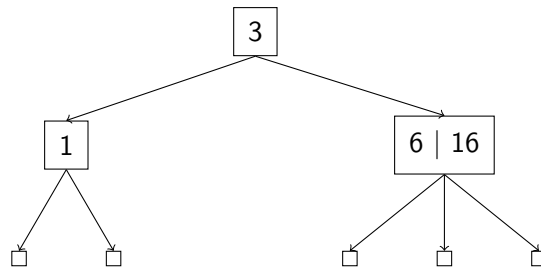
- Fügen Sie die Schlüssel $[10, 3, 1, 16, 5, 4, 7, 6]$ nacheinander ein.
- Löschen Sie die Knoten mit den Schlüsseln $[5, 10, 7, 4]$.
- Zeichnen Sie den korrespondierenden 2,4-Baum.

Lösung:









Aufgabe 3 (2 Punkte): Huffman Coding

Ein gieriger Algorithmus für die Konstruktion von Huffman Coding Bäumen wurde in der Vorlesung gezeigt. Betrachten Sie die folgenden Zeichenhäufigkeiten:

Zeichen	u	a	n	z	o	b
Frequenz	5	3	3	2	1	1

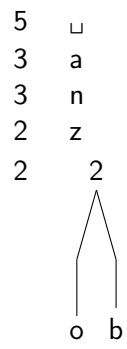
- Erstellen Sie den entsprechenden Huffman Coding Baum. Zeigen Sie jeden Zwischenschritt.
- Zeigen Sie die Codierung und Decodierung mit dem Beispielwort *banana*.

Lösung:

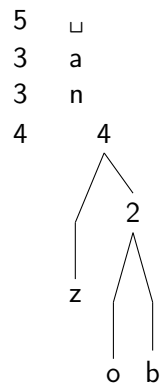
- Eine mögliche Lösung:

—	
5	u
3	a
3	n
2	z
1	o
1	b

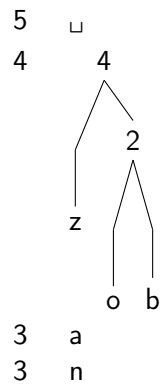
—



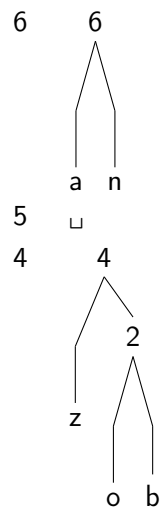
—



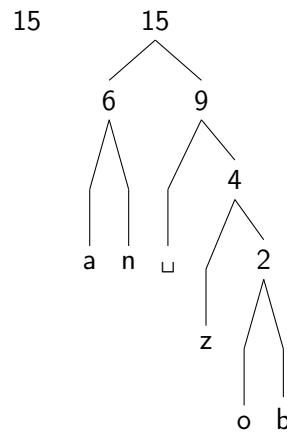
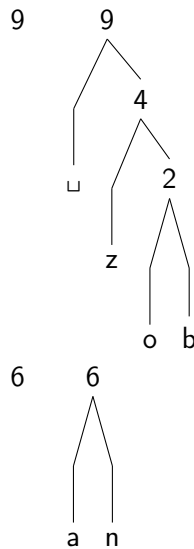
—



—



—



- Codierung:

Finden Sie einen Pfad von der Wurzen zu allen Zeichen im Wort. Erzeugen Sie dabei bei Abbiegung nach links eine 0 und bei Abbiegung nach rechts eine 1.

b 1111 (rechts, rechts, rechts, rechts)

a 00

n 01

1111 00 01 00 01 00 (Die Leerzeichen dienen der Lesbarkeit und sind nicht Teil des Codes.)

Decodierung:

Starten Sie bei der Wurzel und folgen Sie dem Baum den Code entsprechend (0 -> links, 1 -> rechts) nach unten. Wenn ein Blatt erreicht wird, geben Sie das enthaltene Zeichen aus und springen Sie zurück zur Wurzel.

Zum Beispiel: 1111 am Anfang führt bis zum Blatt, das *b* beinhaltet. Nach dem Rücksprung zur Wurzel führt 00 zum Blatt, das *a* beinhaltet und so weiter.

Aufgabe 4 (2 Punkte): Münzrückgabe

Implementieren Sie ein System zur Rückgeldberechnung. Dabei erhalten sie als Eingabe die Kosten eines Produktes x und das dafür erhaltenen Geld y . Zusätzlich gibt es eine Konstante S , die alle Münzdenominationen, die zur Verfügung stehen, enthält. Dabei steht pro Denomination eine unendliche Anzahl von Münzen zur Verfügung.

- Implementieren Sie das System in einer geeigneten Programmiersprache ihrer Wahl. Verwenden sie dafür einen gierigen Algorithmus und testen Sie Ihre Lösung mit $S = \{1, 5, 10\}$, $x = 1, y = 10$.
- Findet Ihr gieriger Algorithmus immer die beste Lösung, wenn Sie S , x und y variieren? Wenn ja, beweisen Sie das, wenn nein, geben Sie ein Gegenbeispiel an.

Lösung:

- Siehe `coin_return.py`.
- Nein, Gegenbeispiel: $S = \{1, 20, 50\}$, $x = 0, y = 60$ (optimale Lösung: $(20, 20, 20)$, gefundene Lösung: $(50, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)$)