Kapitel 1

a) Beschreiben Sie die binäre Suche in eigenen Worten.

b) Nennen Sie die Vorteile der binären gegenüber der linearen Suche.

c) Wenden Sie die binäre Suche auf folgende Liste an, um »Rentier« zu finden.

1. Affe
2. Bär
3. Chamäleon
4. Delfin
5. Erdmännchen
6. Flunder
7. Giraffe
8. Hund
9. Igel
10. Jaguar
11. Katze
12. Luchs
13. Maus
14. Nasenbär
15. Ozelot
16. Papagei
17. Qualle
18. Rentier
19. Schwein
20. Tintenfisch
21. Uhu
22. Vogelspinne
23. Wegschnecke
24. Xenops (ein Singvogel)
25. Yak
26. Zebra

Kapitel 2

a) Beschreiben Sie in eigenen Worten, woraus ein binärer Suchbaum besteht und zeichnen Sie ein Beispiel mit 10 Knoten auf. Die Knoten sollen dabei ausschließlich Zahlen enthalten.

b) Beschreiben Sie anhand Ihres Beispiels den »In order«-Durchlauf und demonstrieren Sie diesen.

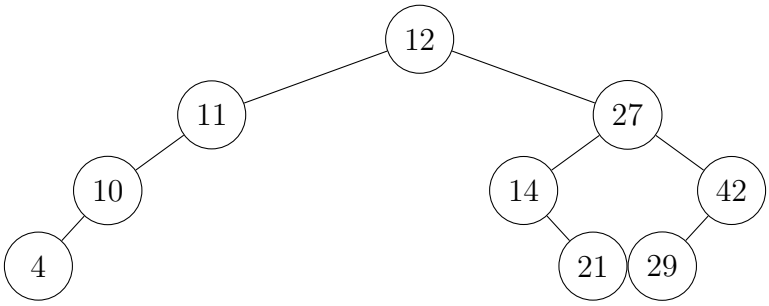
Kapitel 3

a) Beschreiben Sie, welche Attribute und Methoden Objekte vom Typ KNOTEN benötigen.

b) Wie lässt sich der Aufbau eines binären Suchbaums objektorientiert realisieren?

c) Beschreiben Sie in eigenen Worten, wie beim Einfügen eines neuen Knotens vorzugehen ist.

Kapitel 4



a) Geben Sie die Höhendifferenzen für alle Knoten an. Bestimmen Sie die Stelle an welcher die Ausgeglichenheitsbedingung verletzt wird.

b) Welche Knoten ist als letztes dem Baum hinzugefügt worden, so dass dieser nicht mehr balanciert ist.

c) Welche Rotation muss durchgeführt werden, um den Baum auszugleichen? Führen Sie diese Rotation aus.

Lösungen zu den Aufgaben

Lösung 1

a) Zu nennende Kriterien: Aufteilen der Listen in zwei gleiche Teile, danach »Springen« zur Mitte des verbleibenden Teils. Dieser wird betrachtet und von dort in einem Teil rekursiv den Vorgang wiederholen, bis die Mitte das gesuchte Element ist.

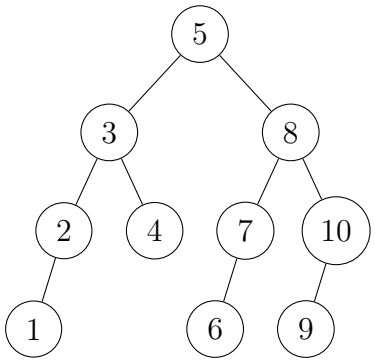
b) Erwartete Antwort: Die Liste muss nicht komplett durchlaufen werden. Sie ist v. a. schnell, um herauszufinden, ob ein Element überhaupt vorhanden ist.

c) Lösungschritte: 13, 20, 16, 18

Lösung 2

a) Zu nennende Elemente: Wurzel, Kanten, Knoten, Teilbäume. Wichtig beim aufgezeichneten Beispiel: Die Eigenschaft aus Abschnitt »Eigenschaften des binären Suchbaums« muss erfüllt sein, d. h. alle Knoten links eines vorgegebenen Knotens müssen kleiner sein als der Knoten selbst.

Ein möglicher Baum sieht so aus:



b) Es wird für jeden Teilbaum immer zuerst der linke Teilbaum traversiert, dann die Mitte ausgegeben und dann der rechte Teilbaum traversiert. Für den oben dargestellten Baum würde das zum Beispiel bedeuten, dass der linke Teilbaum zu 1, 2, 3 und 4 traversiert würde. Dann würde die 5 ausgegeben und dann der rechte Teilbaum traversiert zu 6, 7, 8, 9 und 10.

Lösung 3

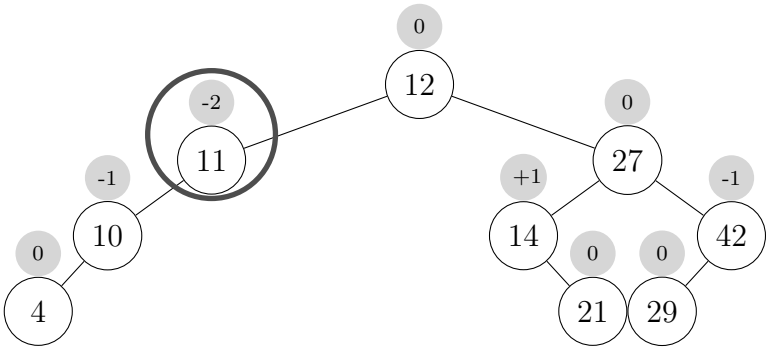
a) Zu nennen sind hier: linker/rechter Nachfolger, ein Attribut zum Speichern eines Wertes; Ausgabe des linken/rechten Nachfolgers; setzen derselben

b) Erwartete Antwort: kennt-Beziehung zwischen den Knoten, so dass die Knoten »aneinander hängen«.

c) Erwartete Antwort: Betrachte Wurzel; Entscheidung für »links« oder »rechts«; wenn kein Nachfolger dort vorhanden, füge ein; sonst wiederhole Vorgehensweise.

Lösung 4

a) Siehe Baum



b) Als letztes wurde die 4 dem Baum hinzugefügt.

c) Erwartete Antwort: Eine R-Rotation um den Knoten 11.

