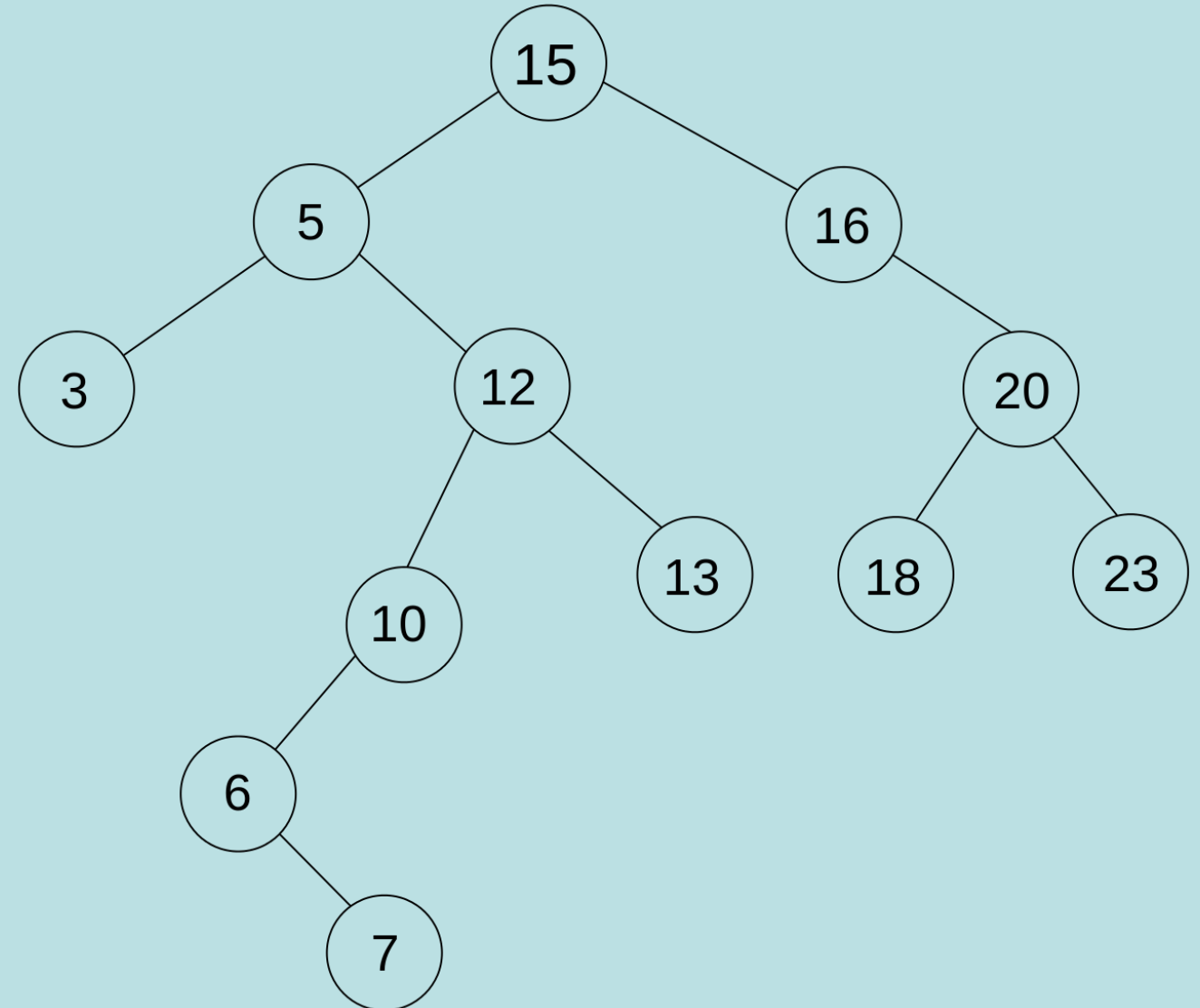


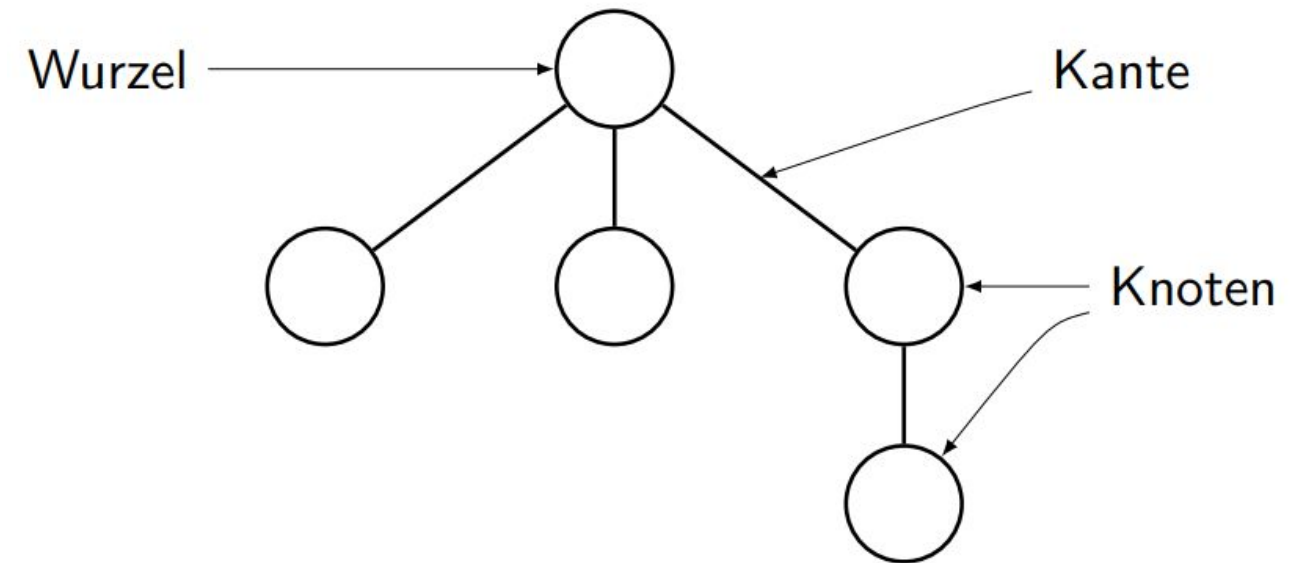
04 Bäume - Teil 1

Algorithmen und Datenstrukturen 2

Bäume

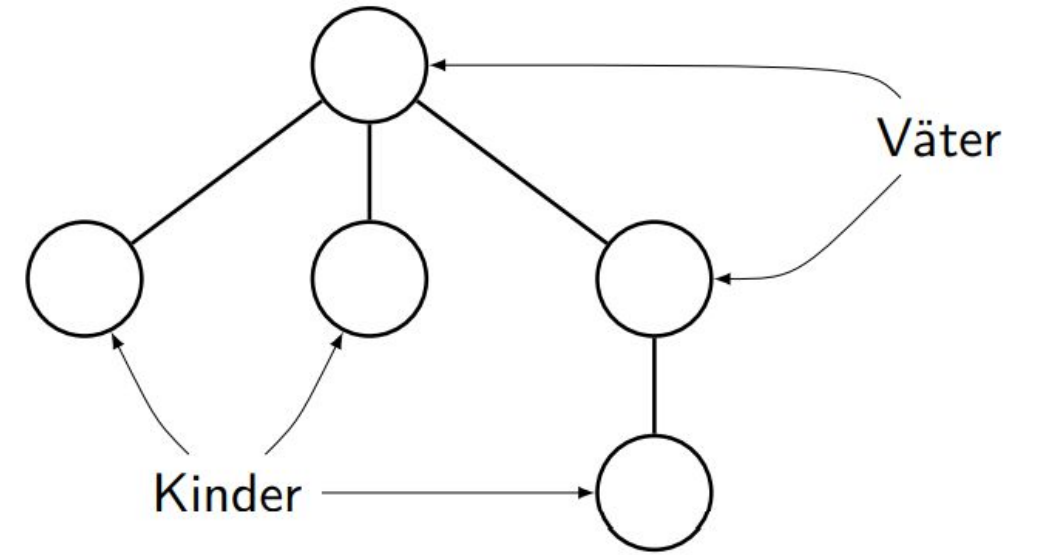


Bäume - Begriffe und Definitionen



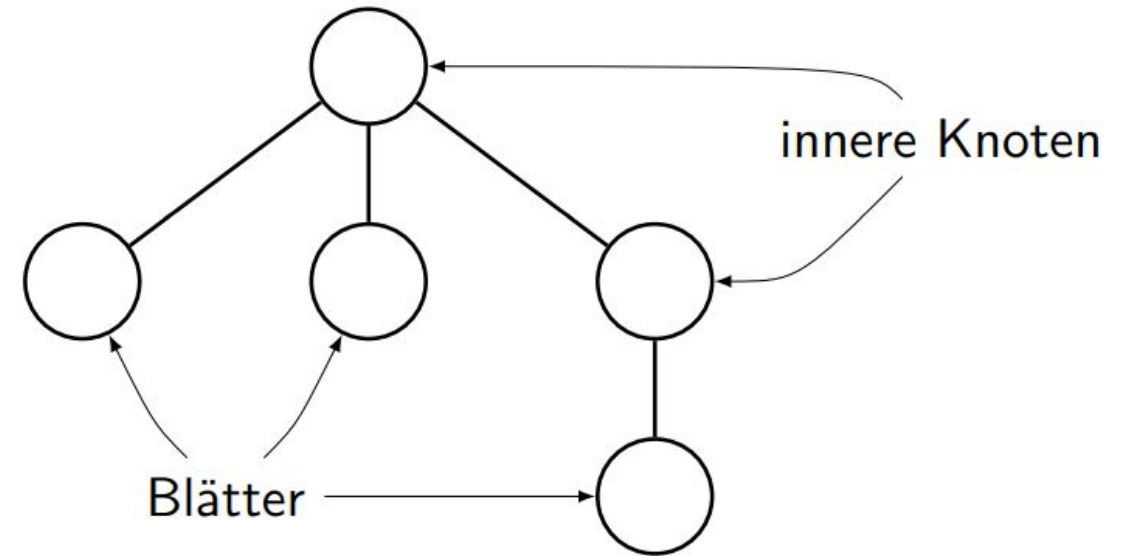
Knoten (nodes) u. Kanten (edges)	Die Knoten dienen zur Speicherung der Daten und sind miteinander durch Kanten verbunden.
Leerer Baum	Ein leerer Baum besitzt keine Knoten (und auch keine Kanten).
Wurzel (root)	Dieser Knoten ist der oberste Knoten in jedem nicht leeren Baum.
Teilbaum	Jeder Knoten im Baum ist seinerseits Wurzel eines (Teil)baums.

Bäume - Begriffe und Definitionen



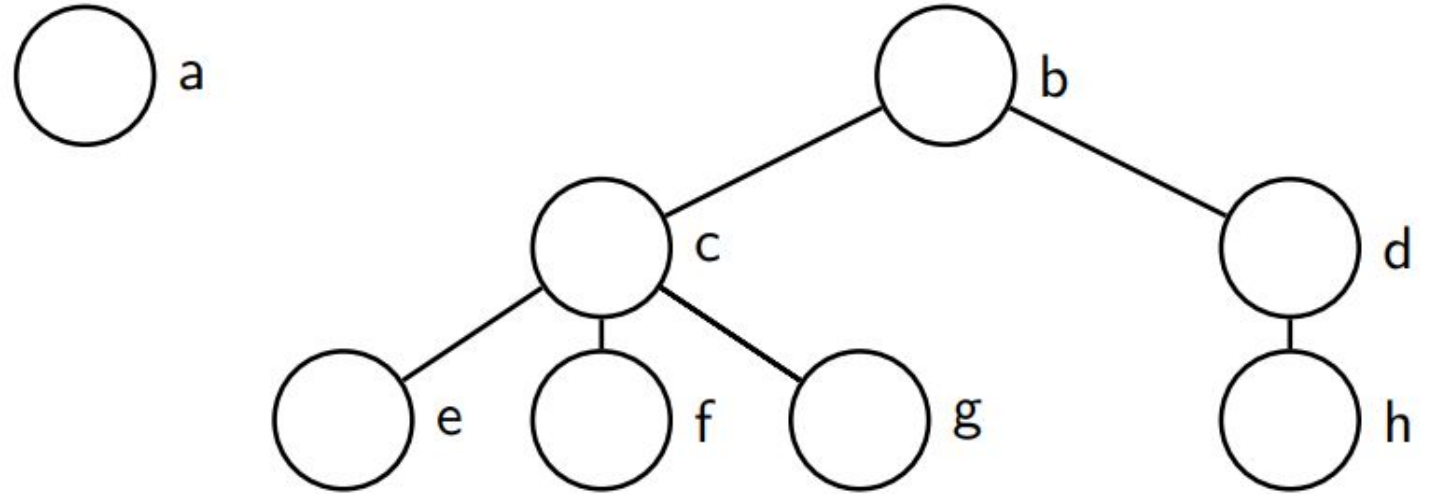
Vater (parent) u. Kind (child)	Väter (oben) sind mit ihren Kindern (unten) durch eine Kante verbunden.
Bruder/Geschwister	Knoten mit dem gleichen Vater.

Bäume - Begriffe und Definitionen



Innerer Knoten (inner node)	Dieser Knoten hat mindestens ein Kind.
Blatt (leaf)	Dieser Knoten hat keine Kinder.

Bäume - Begriffe und Definitionen



Aufgabe:

Bestimmen Sie den Typ aller Knoten in obiger Abbildung.

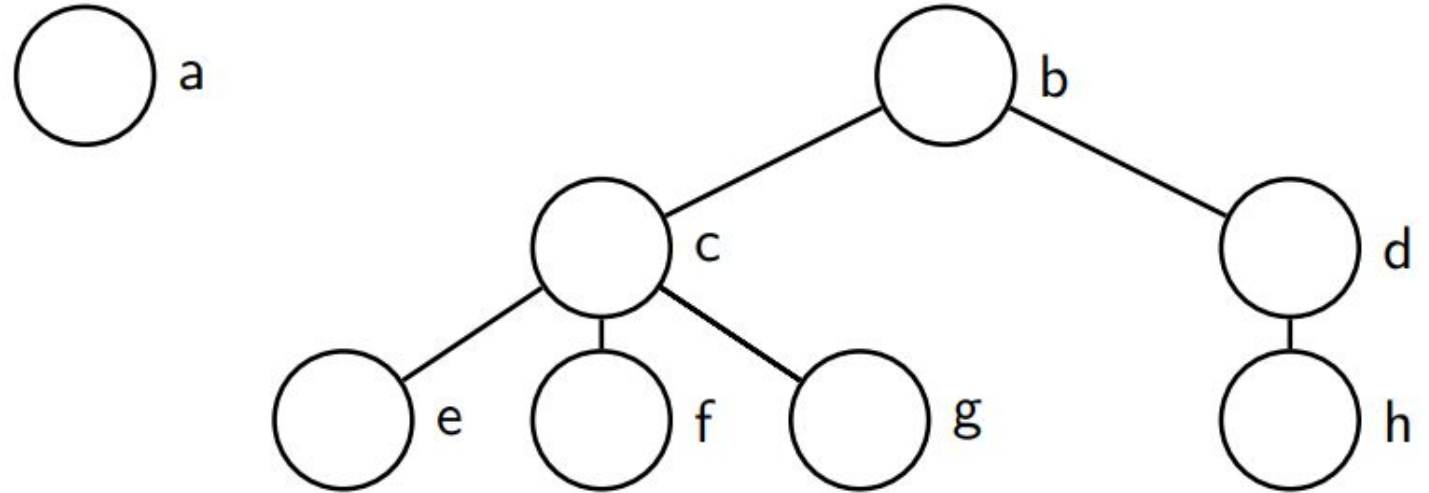
Beachten Sie, dass einzelne Knoten auch von mehreren Typen sein können.

Wurzel

Blatt

Innerer Knoten

Bäume - Begriffe und Definitionen



Aufgabe:

Bestimmen Sie den Typ aller Knoten in obiger Abbildung.

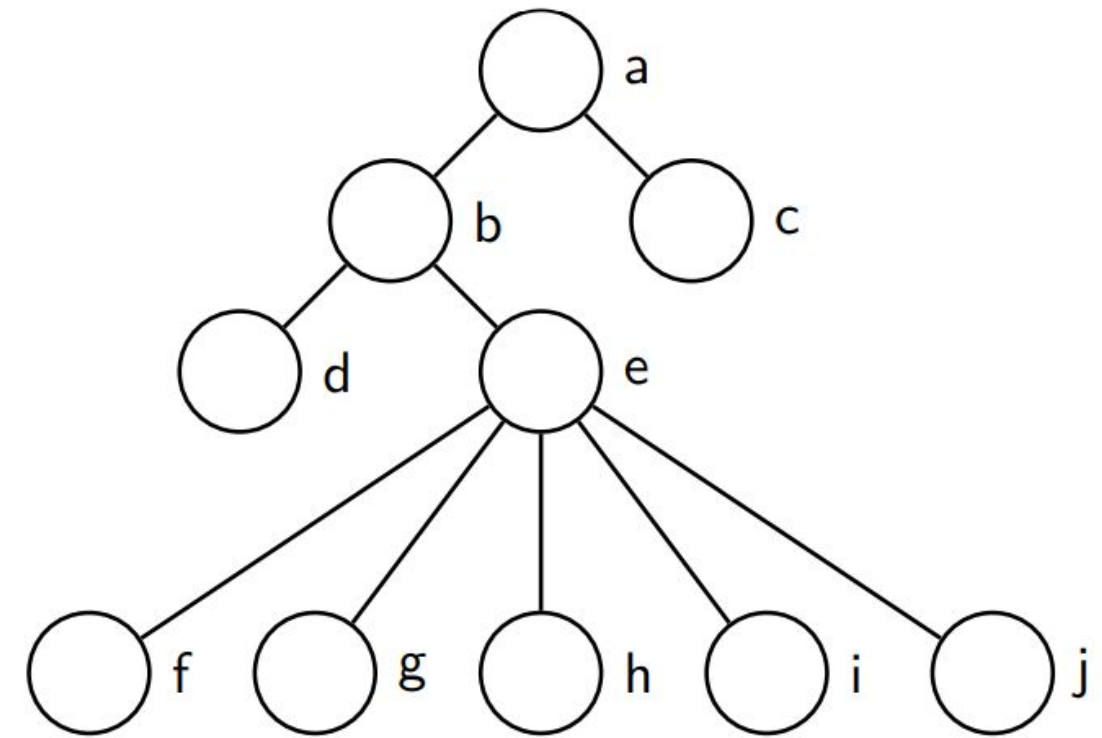
Beachten Sie, dass einzelne Knoten auch von mehreren Typen sein können.

Wurzel Knoten a und b.

Blatt Knoten a, e, f, g und h sind Blätter, da sie keine Kinder haben.

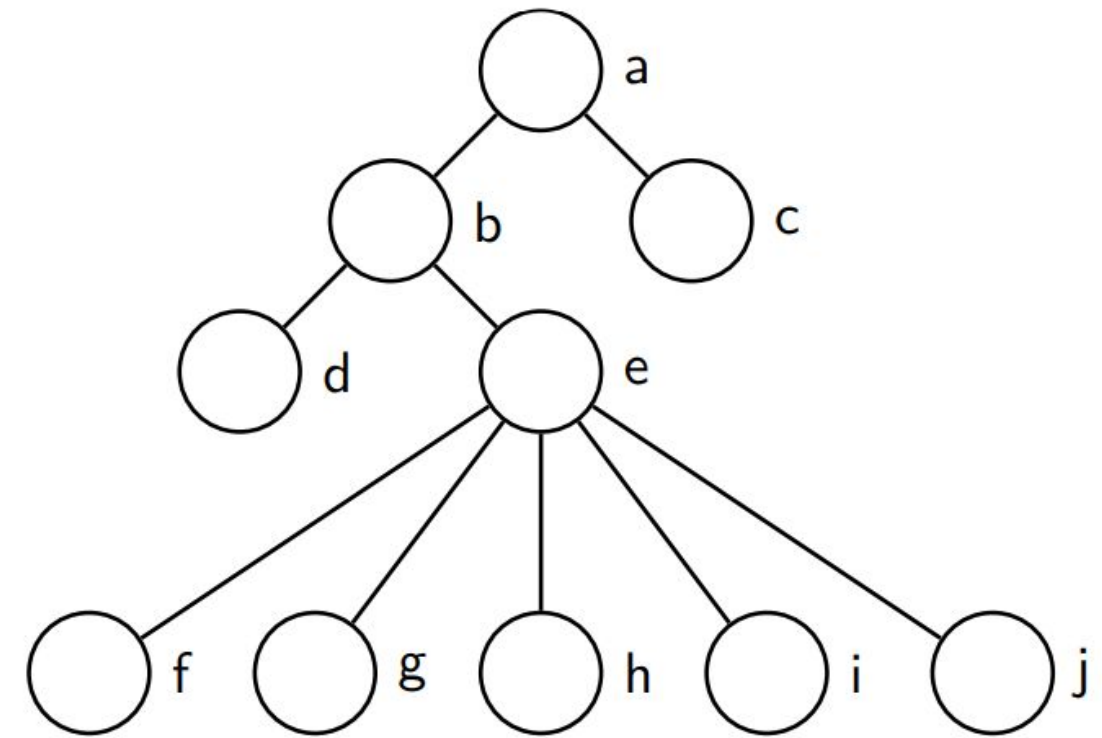
Innerer Knoten Knoten b, c und d, da sie mindestens ein Kind haben

Bäume - Begriffe und Definitionen



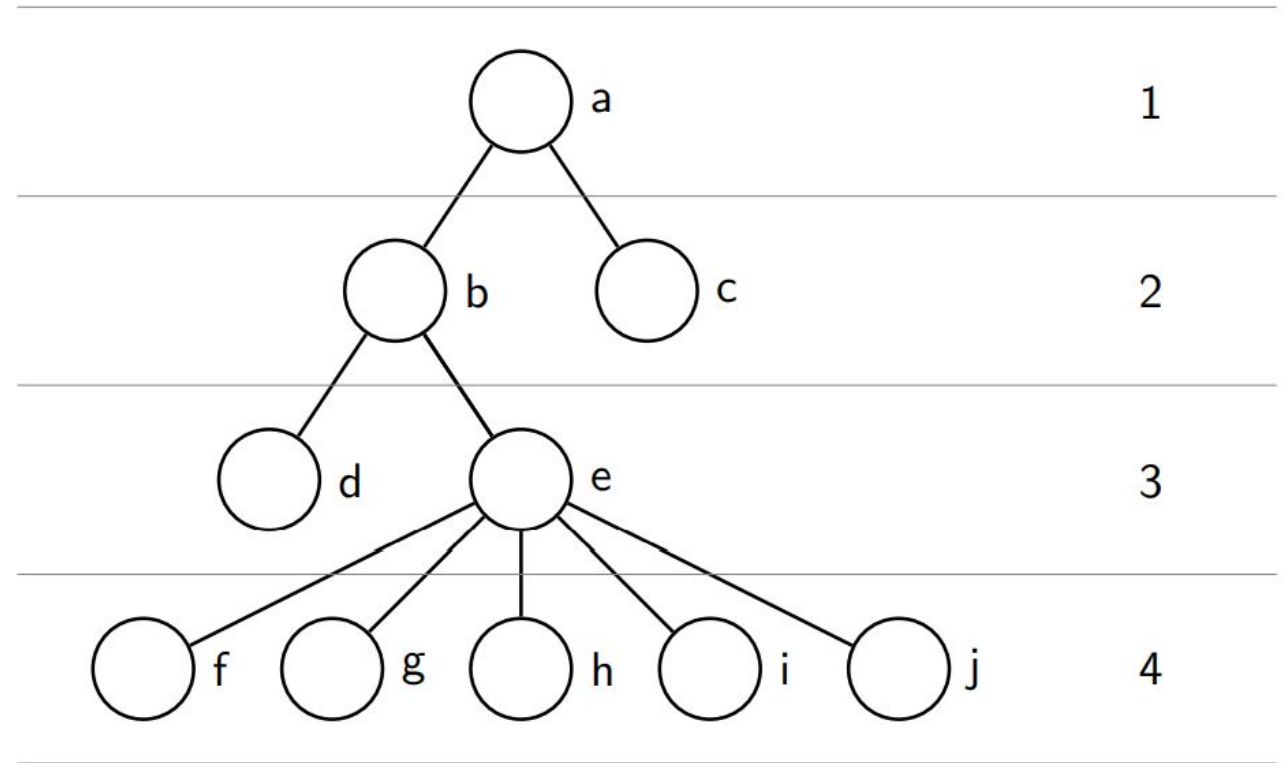
Ordnung (order) des Baumes	<p>Die maximale Anzahl der Kinder, die ein Knoten haben darf.</p> <p><i>An Hand einer Grafik kann nur die Ordnung bestimmt werden, welche der Baum mindestens hat. (Abbildung: Ein Baum mit Ordnung ≥ 5)</i></p>
Grad (degree) des Knotens	<p>Die effektive Anzahl der Kinder, die ein Knoten hat.</p> <p><i>$\text{deg}(v)$ = Anzahl der Kinder von v</i></p> <p><i>Der Grad jedes Knotens muss kleiner oder gleich der Ordnung des Baumes sein. (Abbildung: a hat Grad 2; c hat Grad 0; e hat Grad 5)</i></p>

Bäume - Begriffe und Definitionen



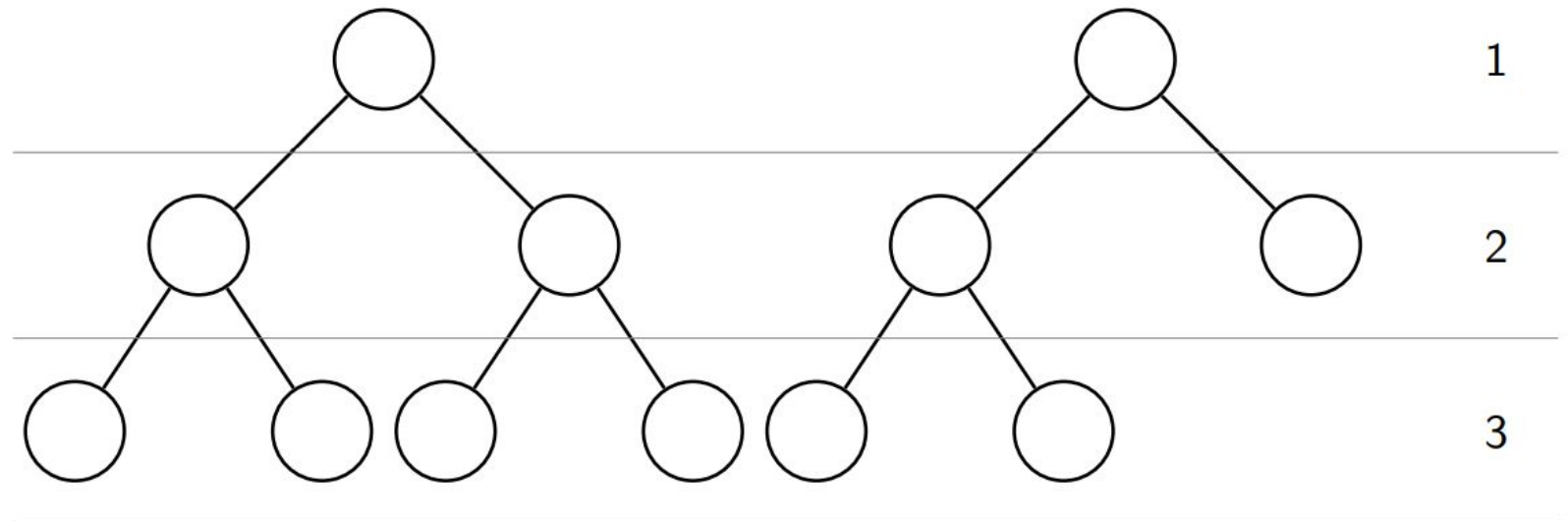
Pfad	Der Weg zwischen zwei Knoten.
Länge eines Pfads	Die Anzahl Knoten auf dem Pfad. <i>Es werden alle Knoten gezählt, welche auf dem Pfad liegen – inklusive des Start- und Endknotens. (Abbildung: Länge des Pfads von h zu b ist 3)</i>

Bäume - Begriffe und Definitionen



Höhe des Baumes	Die Anzahl Knoten auf dem längsten Pfad von der Wurzel in ein Blatt.
Tiefe des Knotens	<p>Die Anzahl Knoten auf dem Pfad zur Wurzel.</p> <p><i>depth(v) = Anzahl der Knoten auf dem Pfad von der Wurzel zum Knoten v</i></p> <p><i>(Abbildung: depth(a) = 1; depth(d) = 3; depth(g) = 4)</i></p>
Niveau / Stufe / Level	Die Knoten der gleichen Tiefe werden zu Niveaus zusammengefasst.

Bäume - Begriffe und Definitionen



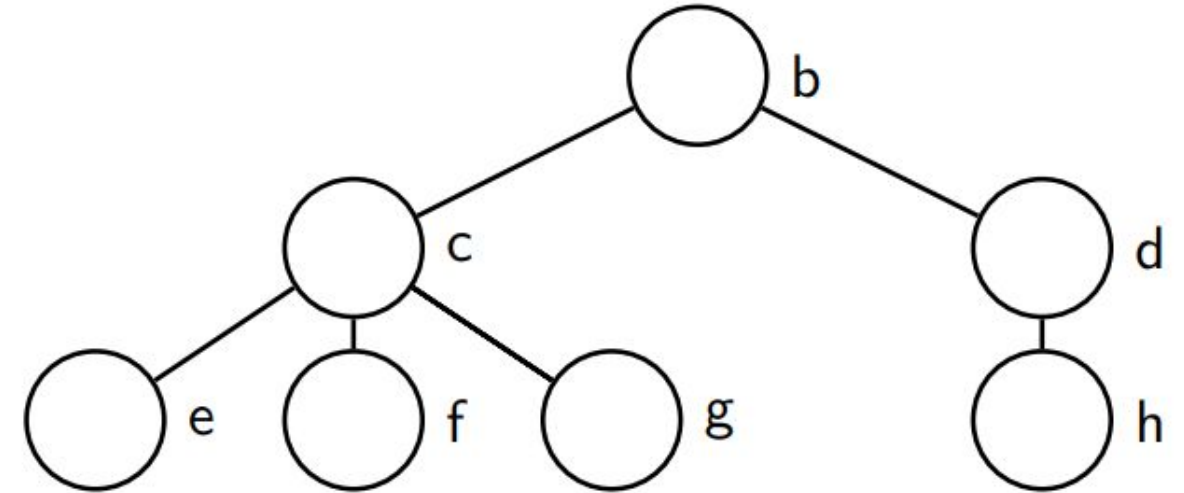
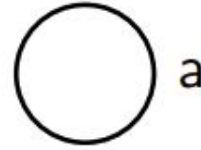
Ausgefüllter Baum	<p>Jeder innere Knoten besitzt die maximale Anzahl Kinder.</p> <p><i>(Abbildung: Wenn die Ordnung 2 ist, dann sind beide Bäume ausgefüllt.)</i></p>
Vollständiger Baum	<p>Jedes Niveau hat die maximale Anzahl Knoten.</p> <p><i>(Abbildung: Für Ordnung 2 ist nur der linke Baum vollständig.)</i></p>

Bäume - Begriffe und Definitionen

Aufgabe:

Bestimmen Sie für die beiden Bäume in der Abbildung:

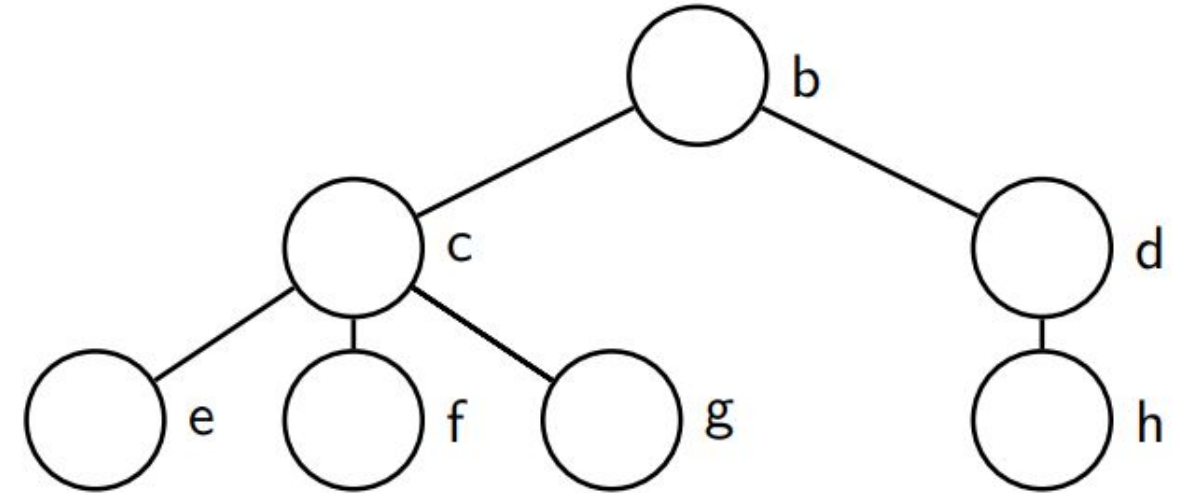
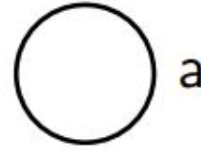
- A. die Höhe
- B. die Ordnung
- C. ob der Baum ausgefüllt ist
- D. den Grad aller Knoten
- E. die Tiefe aller Knoten



Bäume - Begriffe und Definitionen

Aufgabe:

Bestimmen Sie für die beiden Bäume in der Abbildung:



- A. die Höhe
- B. die Ordnung
- C. ob der Baum ausgefüllt ist
- D. den Grad aller Knoten
- E. die Tiefe aller Knoten

Höhe = 1
mindestens 0
ja
 $\deg(a)=0$
 $\text{depth}(a)=1$

Höhe = 3
mindestens 3
nein
 $\deg(b)=2, \deg(c)=3, \deg(d)=1, \deg(e,f,g,h)=0$
 $\text{depth}(b)=1, \text{depth}(c,d)=2, \text{depth}(e,f,g,h)=3$

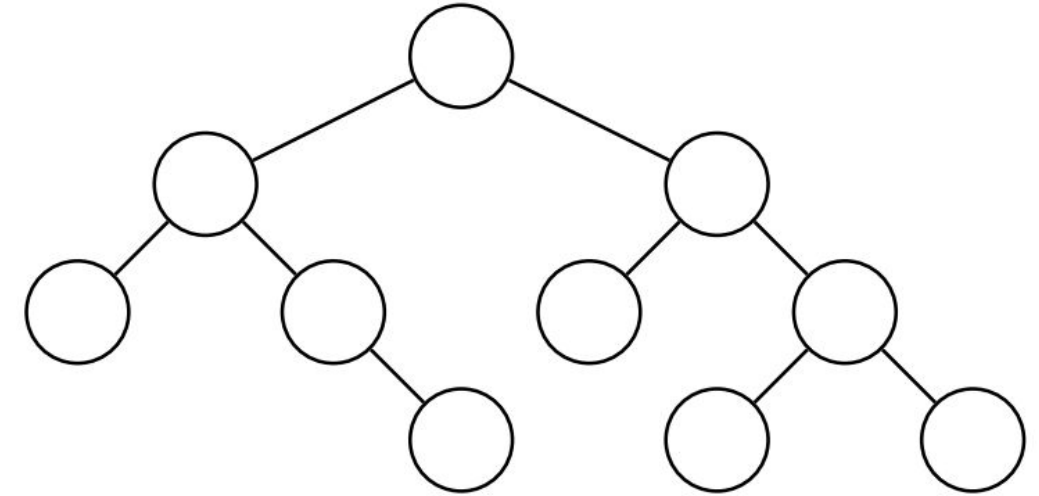
Bäume - Begriffe und Definitionen

Aufgabe:

Arbeitsblatt Aufgaben 1 - 4

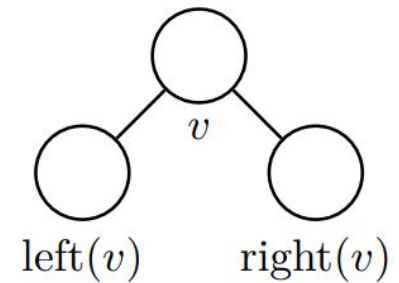
Binärbäume

Binärbaum - Begriffe und Definitionen

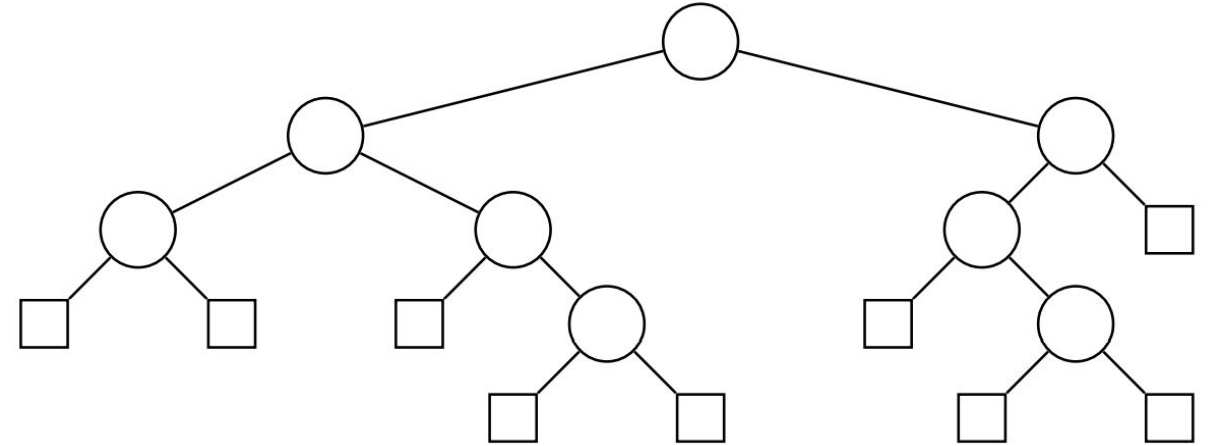


Binärbaum

Ein Baum der Ordnung 2. Knoten sind entweder linkes oder rechtes Kind ihres Elternknotens.



Binärbaum - Begriffe und Definitionen



Binärbaum (alternative Darstellung)	Durch eingefügte leere Blätter ist dieser Binärbaum immer ausgefüllt.
Umwandlung (von Binärbaum B in alternative Variante B')	<ol style="list-style-type: none"> 1. Alle Blätter in B erhalten zwei leere Blätter und werden zu inneren Knoten. 2. Alle inneren Knoten in B, die noch nicht zwei Kinder besitzen, erhalten an entsprechender Stelle ein leeres Blatt.
Vorteil	Diese Variante der Definition erlaubt es gewisse Algorithmen einfacher zu definieren, da alle inneren Knoten genau zwei Kinder haben.
Nachteil	Andererseits werden die Abbildungen dadurch überladen und die Beschreibungen und Definitionen unnötig kompliziert.

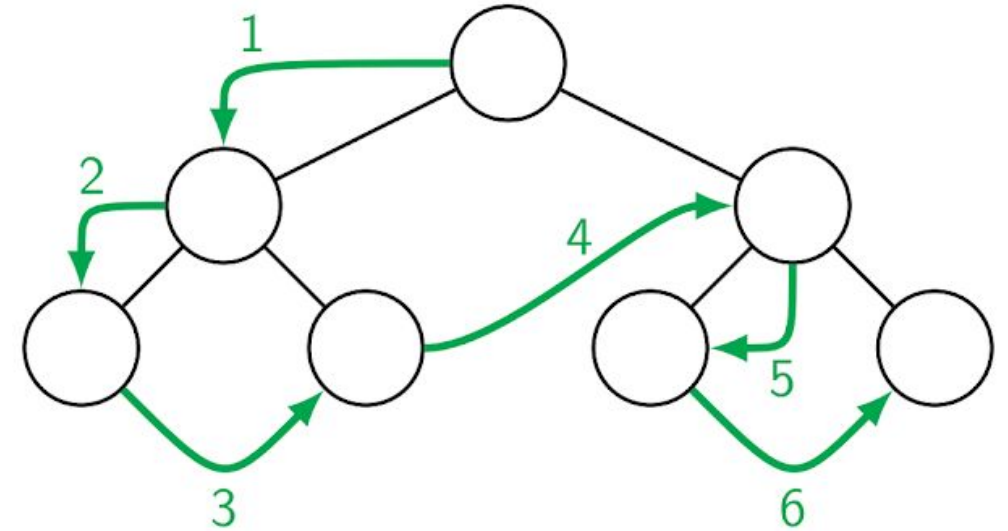
Binärbaum - Traversierungen

Durchlaufordnung	Die Reihenfolge, in der die Knoten eines Baumes durchlaufen werden.
-------------------------	---

<https://studyflix.de/informatik/binarbaum-1362>

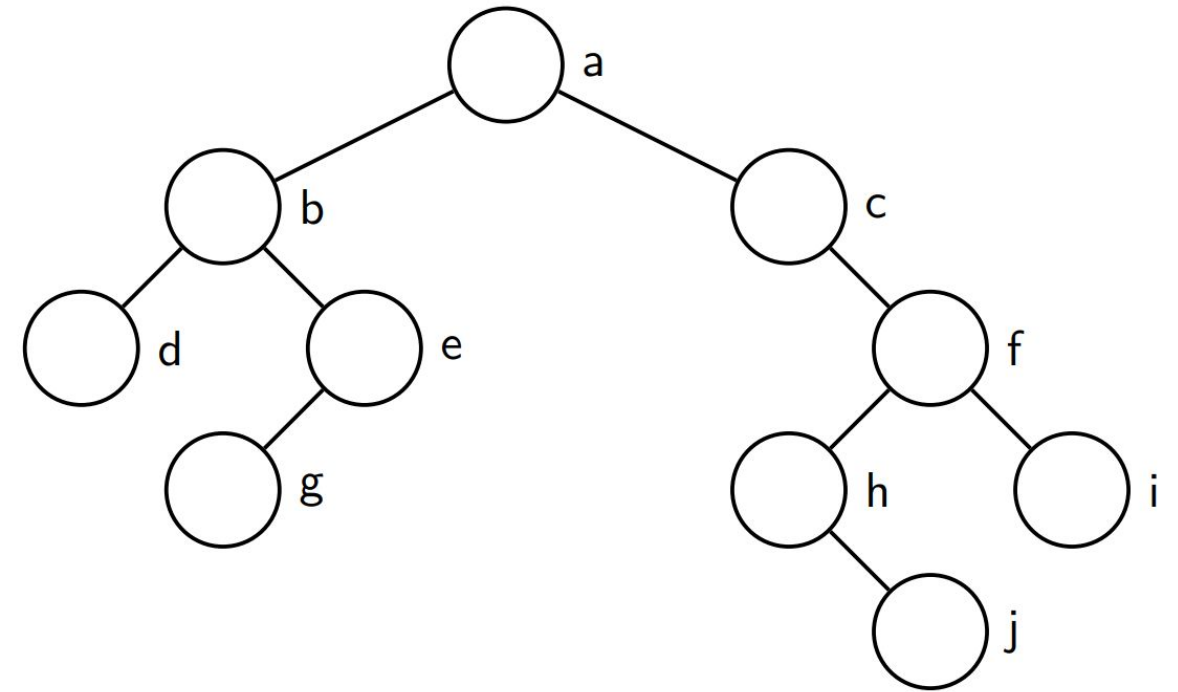
Minute 02:16 bis 04:27

Binärbaum - Traversierungen



Preorder-Reihenfolge (Hauptreihenfolge)	Bei dieser Reihenfolge wird ein Knoten <i>vor</i> seinem linken und rechten Teilbaum durchlaufen.
rekursiver Algorithmus	<p>Durchlaufen eines Binärbaumes mit Wurzel v in der Preorder-Reihenfolge:</p> <ol style="list-style-type: none"> 1. Besuche den Knoten v. 2. Durchlaufe den linken Teilbaum des Knotens v in der Preorder-Reihenfolge. 3. Durchlaufe den rechten Teilbaum des Knotens v in der Preorder-Reihenfolge.

Binärbaum - Traversierungen

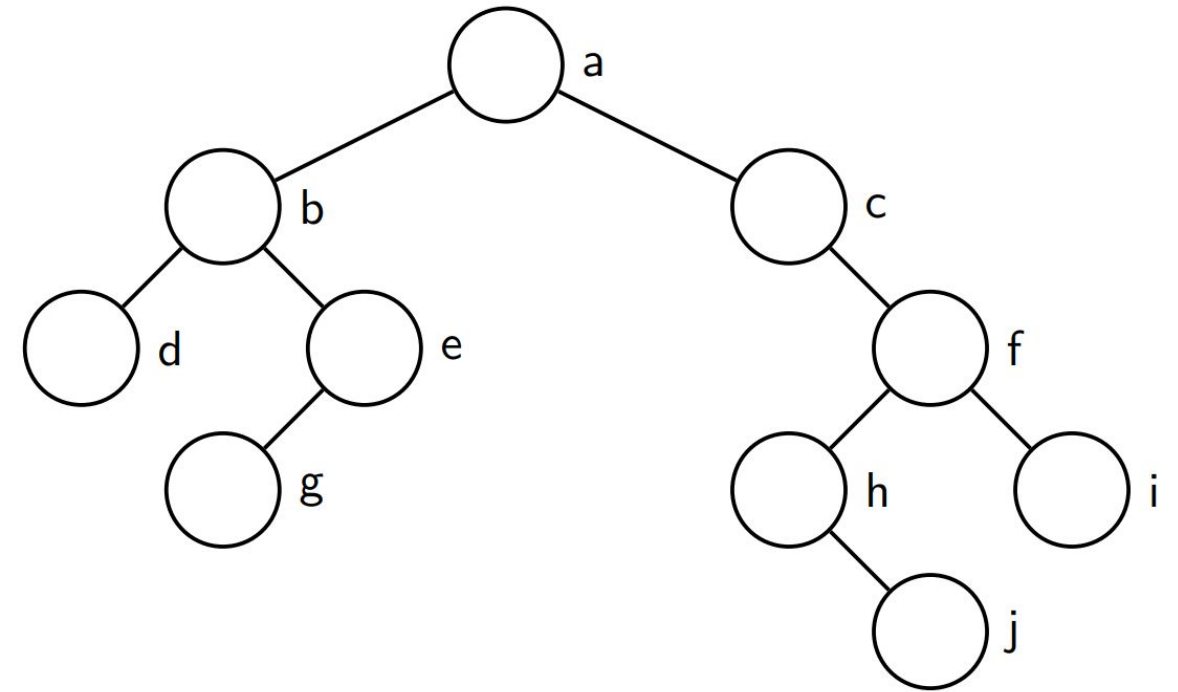


Aufgabe:

Bestimmen Sie die Preorder-Reihenfolge für diesen Binärbaum:

Preorder-Reihenfolge:

Binärbaum - Traversierungen

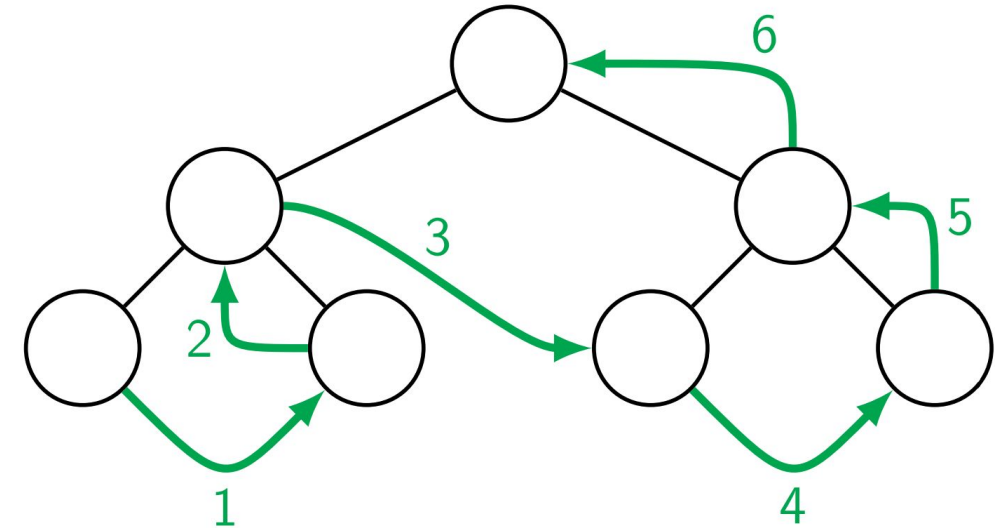


Aufgabe:

Bestimmen Sie die Preorder-Reihenfolge für diesen Binärbaum:

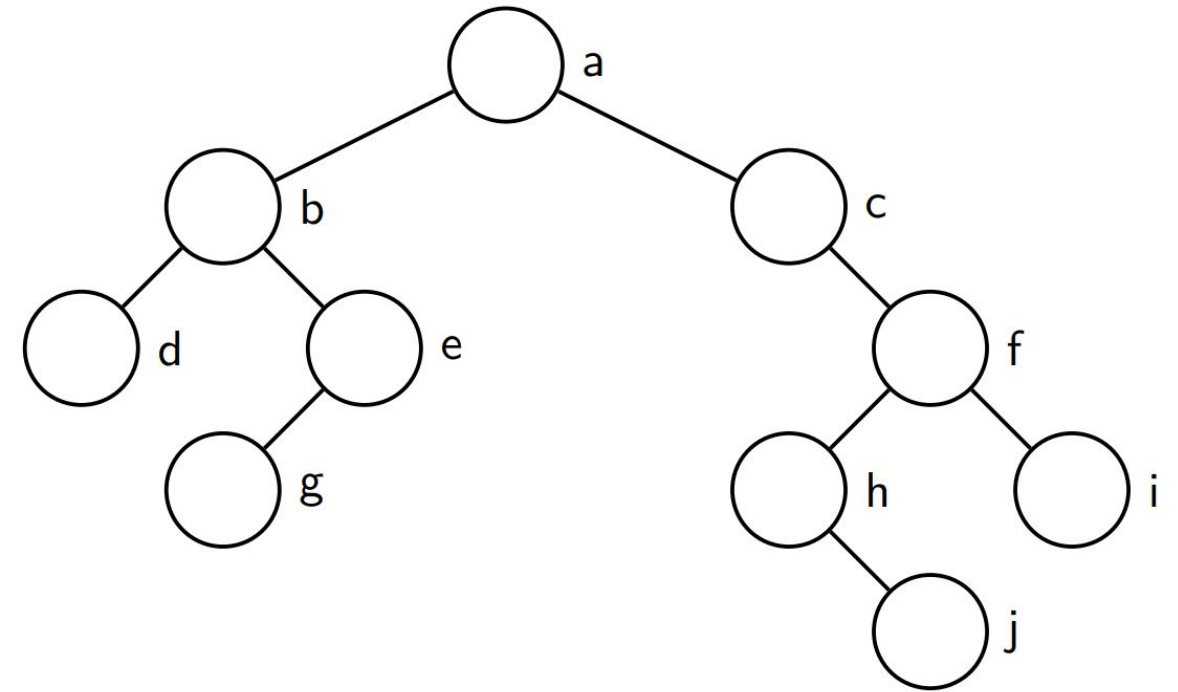
Preorder-Reihenfolge: a, b, d, e, g, c, f, h, j, i

Binärbaum - Traversierungen



Postorder-Reihenfolge (Nebenreihenfolge)	Bei dieser Reihenfolge wird ein Knoten <i>nach</i> seinem linken und rechten Teilbaum durchlaufen.
rekursiver Algorithmus	<p>Durchlaufen eines Binärbaumes mit Wurzel v in der Postorder-Reihenfolge:</p> <ol style="list-style-type: none"> 1. Durchlaufe den linken Teilbaum des Knotens v in der Postorder-Reihenfolge. 2. Durchlaufe den rechten Teilbaum des Knotens v in der Postorder-Reihenfolge. 3. Besuche den Knoten v.

Binärbaum - Traversierungen

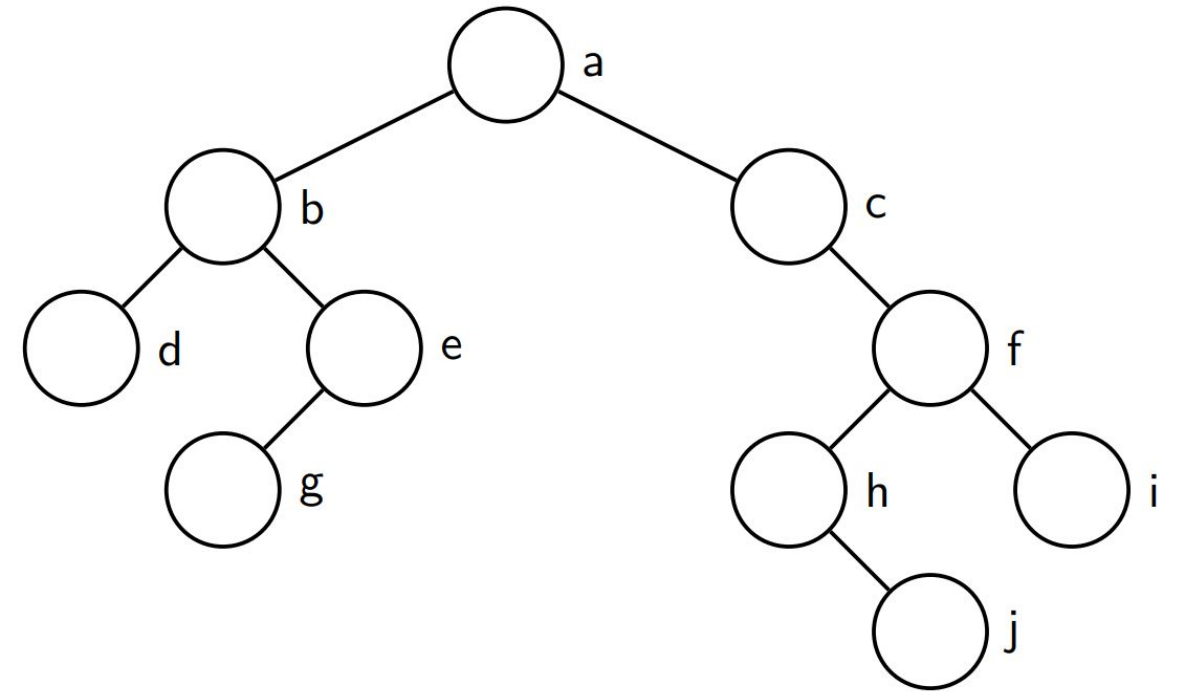


Aufgabe:

Bestimmen Sie die Postorder-Reihenfolge für diesen Binärbaum:

Postorder-Reihenfolge:

Binärbaum - Traversierungen

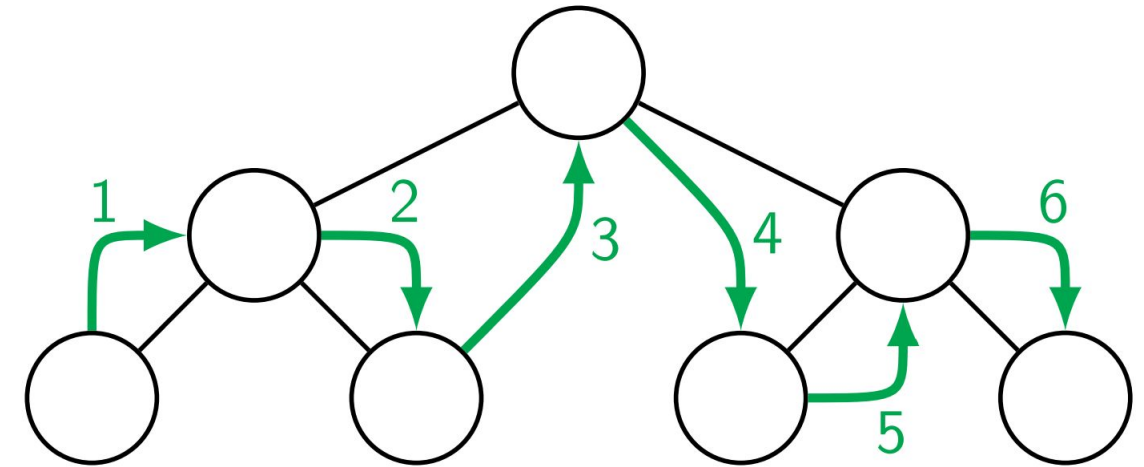


Aufgabe:

Bestimmen Sie die Postorder-Reihenfolge für diesen Binärbaum:

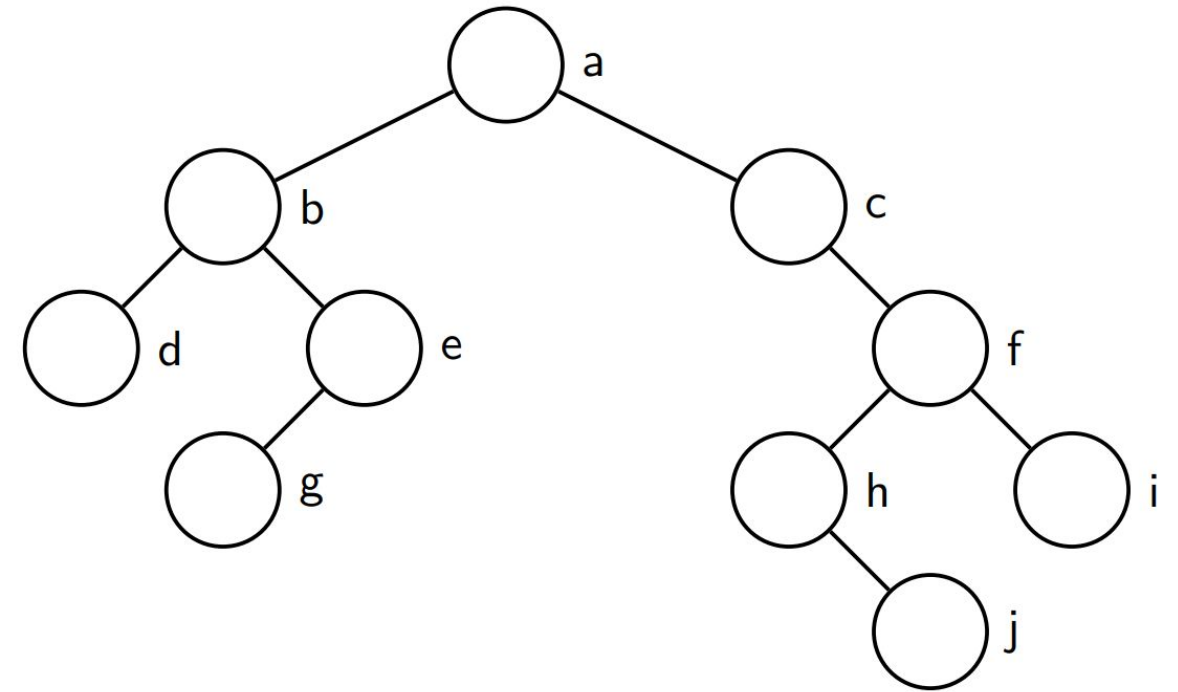
Postorder-Reihenfolge: d, g, e, b, j, h, i, f, c, a

Binärbaum - Traversierungen



Inorder-Reihenfolge (Symmetrische Reihenfolge)	Bei dieser Reihenfolge wird ein Knoten <i>zwischen</i> seinem linken und rechten Teilbaum durchlaufen.
rekursiver Algorithmus	<p>Durchlaufen eines Binärbaumes mit Wurzel v in der Inorder-Reihenfolge:</p> <ol style="list-style-type: none"> 1. Durchlaufe den linken Teilbaum des Knotens v in der Inorder-Reihenfolge. 2. Besuche den Knoten v. 3. Durchlaufe den rechten Teilbaum des Knotens v in der Inorder-Reihenfolge.

Binärbaum - Traversierungen

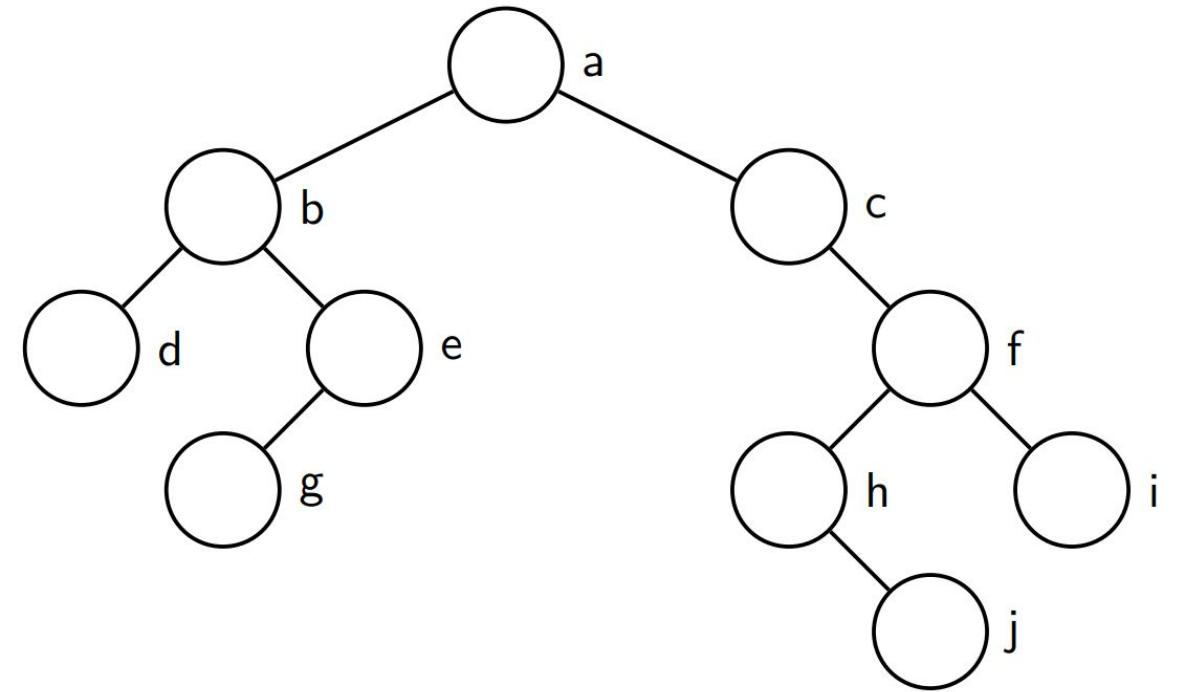


Aufgabe:

Bestimmen Sie die Inorder-Reihenfolge für diesen Binärbaum:

Inorder-Reihenfolge:

Binärbaum - Traversierungen



Aufgabe:

Bestimmen Sie die Inorder-Reihenfolge für diesen Binärbaum:

Inorder-Reihenfolge: d, b, g, e, a, c, h, j, f, i

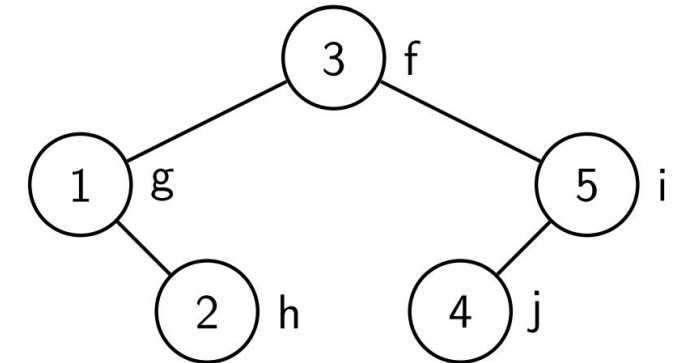
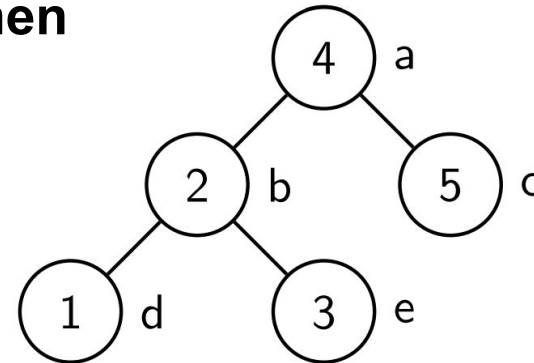
Binärbäume

Aufgabe:

Arbeitsblatt Aufgaben 5 - 8

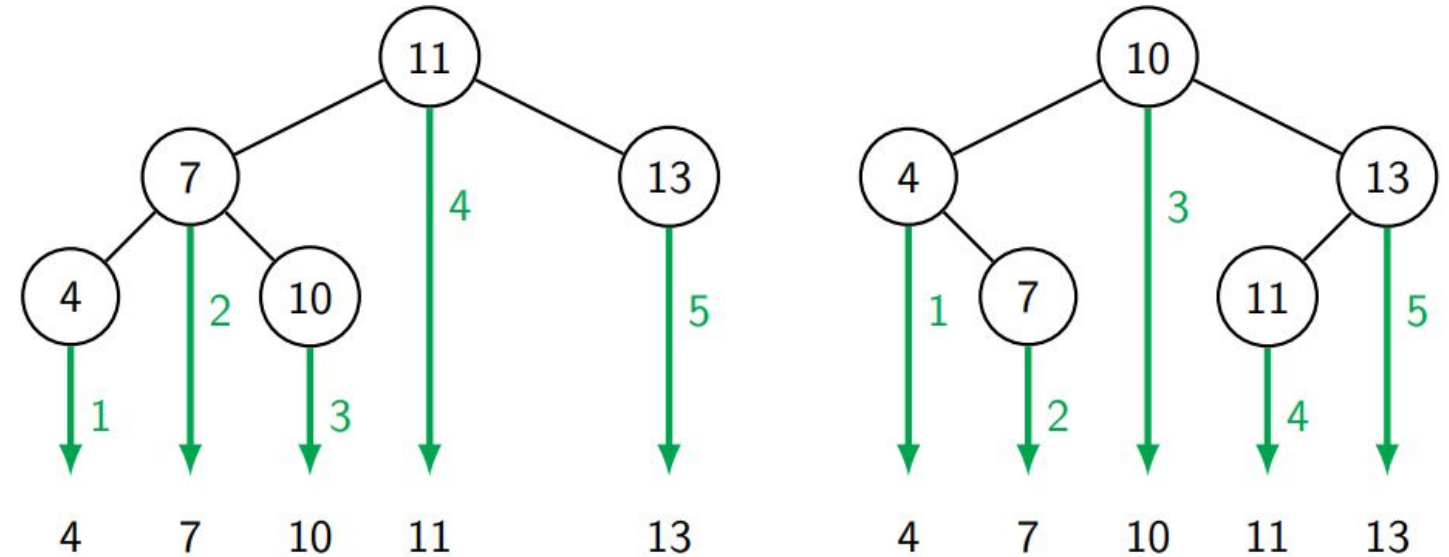
Binäre Suchbäume

Binäre Suchbäume - Begriffe und Definitionen



Binärer Suchbaum (binary search tree)	<p>Ein geordneter Binärbaum, wobei jeder Knoten einen Schlüssel beinhaltet.</p> <ul style="list-style-type: none"> • Jeder Schlüssel im linken Teilbaum eines Knotens ist kleiner als der Schlüssel im Knoten selbst. • Jeder Schlüssel im rechten Teilbaum eines Knotens ist grösser als oder gleich dem Schlüssel im Knoten selbst.
Schlüssel (key)	<p>Der Schlüssel kennzeichnet die Daten die in einem Knoten abgelegt sind.</p> <p>$key(v)$ = Schlüssel von Knoten v</p>

Binäre Suchbäume - Ordnung



Ordnung

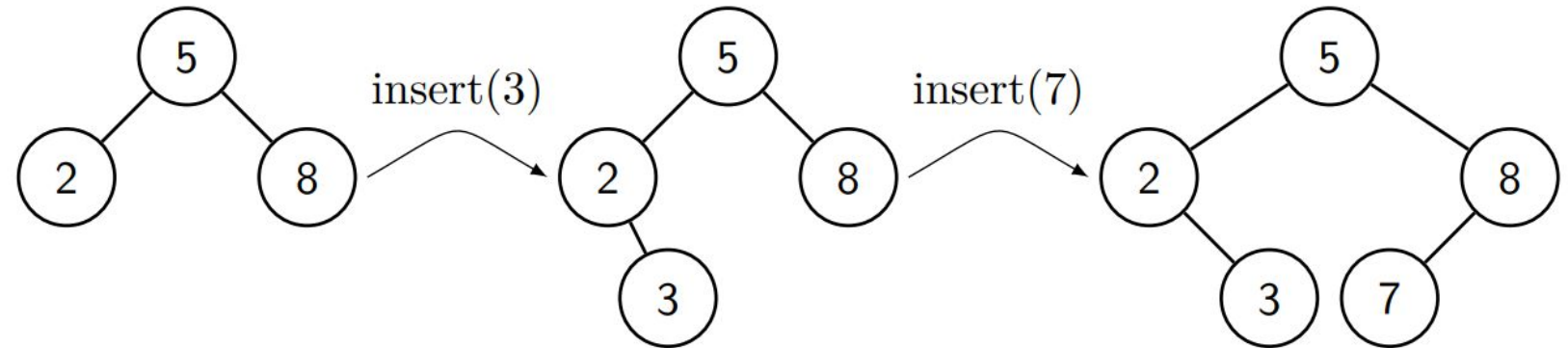
Bei der Inorder-Reihenfolge der Knoten werden die Schlüssel in aufsteigender Reihenfolge betrachtet.

Binäre Suchbäume - Operationen

- Einfügen (insert)
- Suchen (search)
- Entfernen (remove)

<https://studyflix.de/informatik/binaerer-suchbaum-1364>

Binäre Suchbäume - Operationen



Einfügen

Das Einfügen (insert) eines Knotens erfordert das Suchen der richtigen Einfügestelle im Baum.

1. In der Wurzel des Baumes wird eine Vergleichsoperation ausgeführt und entschieden, ob der einzufügende Knoten im linken oder rechten Teilbaum gespeichert werden soll.
2. Dieser Vergleich wird rekursiv ausgeführt bis der jeweilige Teilbaum leer ist. Hier ist nun der freie Platz, an dem der Knoten mit den Daten eingefügt werden kann.

Binäre Suchbäume - Operationen

Aufgabe

Ausgangslage ist jeweils ein leerer Baum. Fügen Sie die folgenden Knoten in den drei verschiedenen vorgegebenen Reihenfolgen nacheinander in den Baum ein.

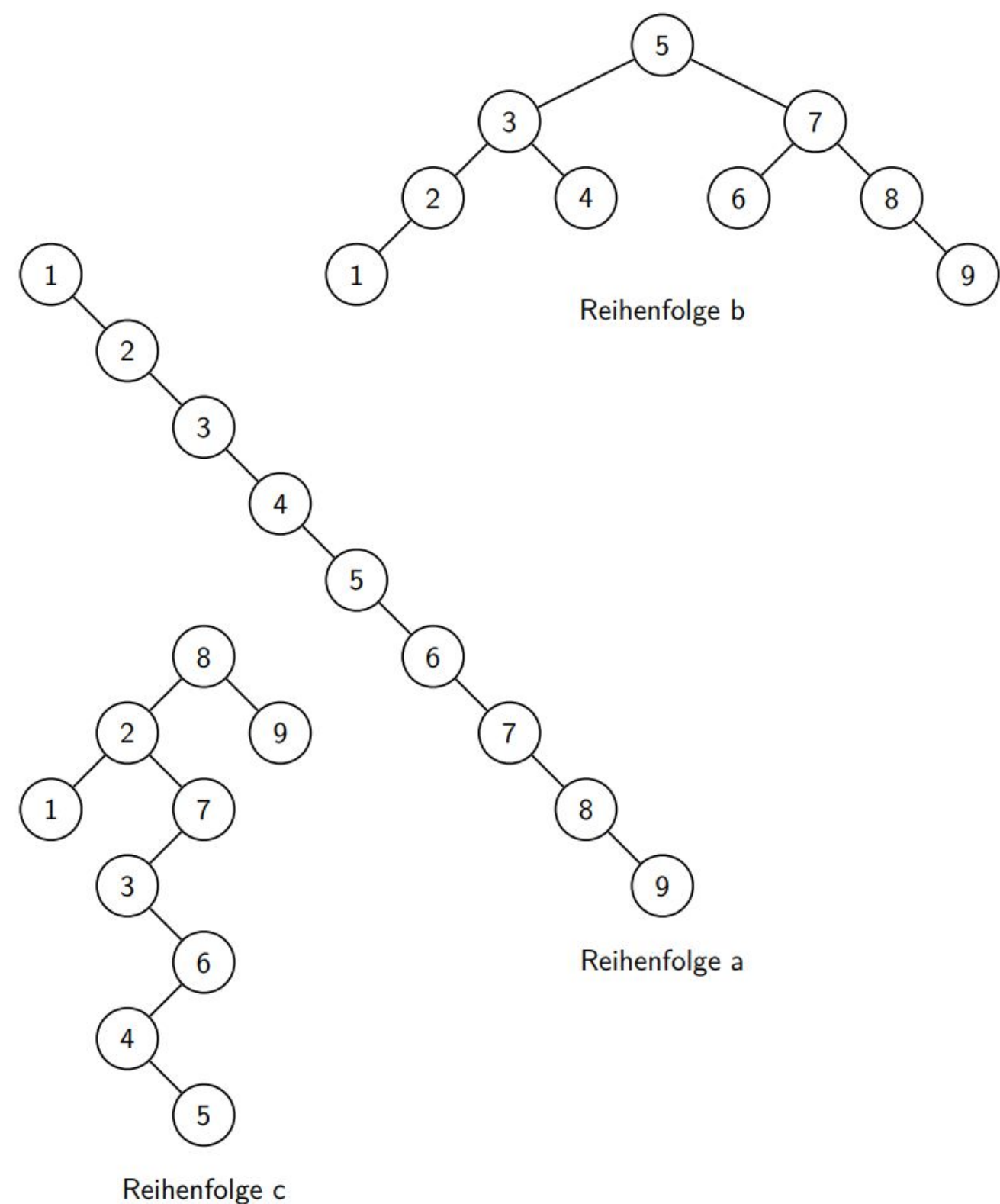
- Reihenfolge a: 1, 2, 3, 4, 5, 6, 7, 8, 9
- Reihenfolge b: 5, 3, 7, 4, 6, 2, 8, 1, 9
- Reihenfolge c: 8, 2, 7, 3, 6, 4, 5, 9, 1

Binäre Suchbäume - Operationen

Aufgabe

Ausgangslage ist jeweils ein leerer Baum. Fügen Sie die folgenden Knoten in den drei verschiedenen vorgegebenen Reihenfolgen nacheinander in den Baum ein.

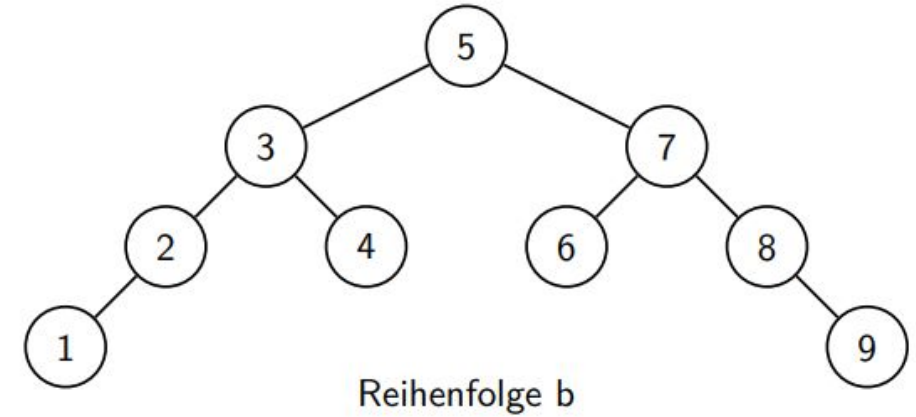
- Reihenfolge a: 1, 2, 3, 4, 5, 6, 7, 8, 9
- Reihenfolge b: 5, 3, 7, 4, 6, 2, 8, 1, 9
- Reihenfolge c: 8, 2, 7, 3, 6, 4, 5, 9, 1



Binäre Suchbäume - Operationen

Aufgabe

Nennen Sie mindestens zwei weitere Reihenfolgen, welche den gleichen binären Suchbaum erzeugen.

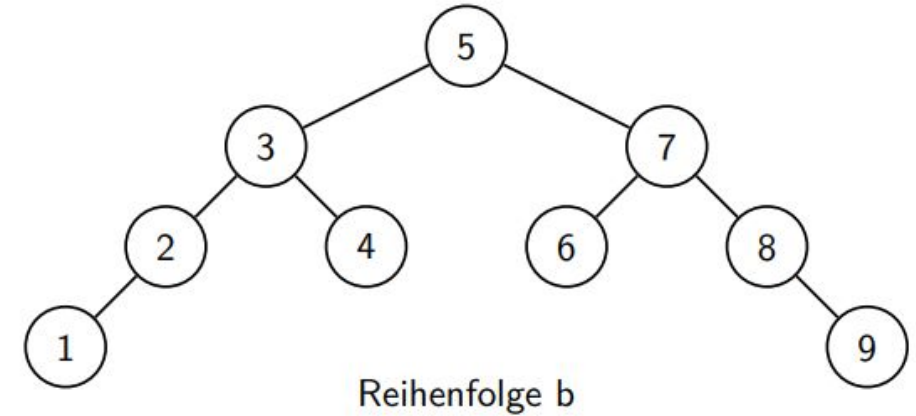


Binäre Suchbäume - Operationen

Aufgabe - Lösung

Es gibt mehrere Reihenfolgen, welche den gleichen binären Suchbaum erzeugen. Zwei mögliche Lösungen sind:

1. Möglichkeit: 5, 3, 4, 2, 1, 7, 6, 8, 9
2. Möglichkeit: 5, 7, 8, 9, 6, 3, 2, 1, 4



Binäre Suchbäume - Operationen

Suchen

Um einen Knoten in einem binären Suchbaum zu suchen (search), muss der zu suchende Schlüssel k bekannt oder berechenbar sein.

1. Der Schlüssel des einzufügenden Knotens wird mit der Wurzel verglichen. Ist er gleich, wurde der Knoten gefunden und die Suche beendet. Ist er kleiner, wird die Suche im linken Teilbaum fortgesetzt. Ist er grösser, wird im rechten Teilbaum weitergesucht.
2. Diese Vergleiche werden rekursiv ausgeführt bis die Suche erfolgreich ist oder erfolglos in einem leeren Teilbaum endet.

Binäre Suchbäume - Operationen

Aufgabe

Wie müssen Sie den Algorithmus abändern, damit alle Knoten mit Schlüssel k gefunden werden?

Suche einen Knoten mit Schlüssel k

- 1. Der Schlüssel des einzufügenden Knotens wird mit der Wurzel verglichen. Ist er gleich, wurde der Knoten gefunden und die Suche beendet. Ist er kleiner, wird die Suche im linken Teilbaum fortgesetzt. Ist er grösser, wird im rechten Teilbaum weitergesucht.*
- 2. Diese Vergleiche werden rekursiv ausgeführt bis die Suche erfolgreich ist oder erfolglos in einem leeren Teilbaum endet.*

Binäre Suchbäume - Operationen

Aufgabe - Lösung

Wie müssen Sie den Algorithmus abändern, damit alle Knoten mit Schlüssel k gefunden werden?

Suche **alle** Knoten mit Schlüssel k

1. Der Schlüssel des einzufügenden Knotens wird mit der Wurzel verglichen. Ist er gleich, wurde **ein** Knoten gefunden und **die Suche im rechten Teilbaum fortgesetzt**. Ist er kleiner, wird die Suche im linken Teilbaum fortgesetzt. Ist er grösser, wird im rechten Teilbaum weitergesucht.
2. Diese Vergleiche werden rekursiv ausgeführt **bis die Suche in einem leeren Teilbaum endet**.

Binäre Suchbäume - Operationen

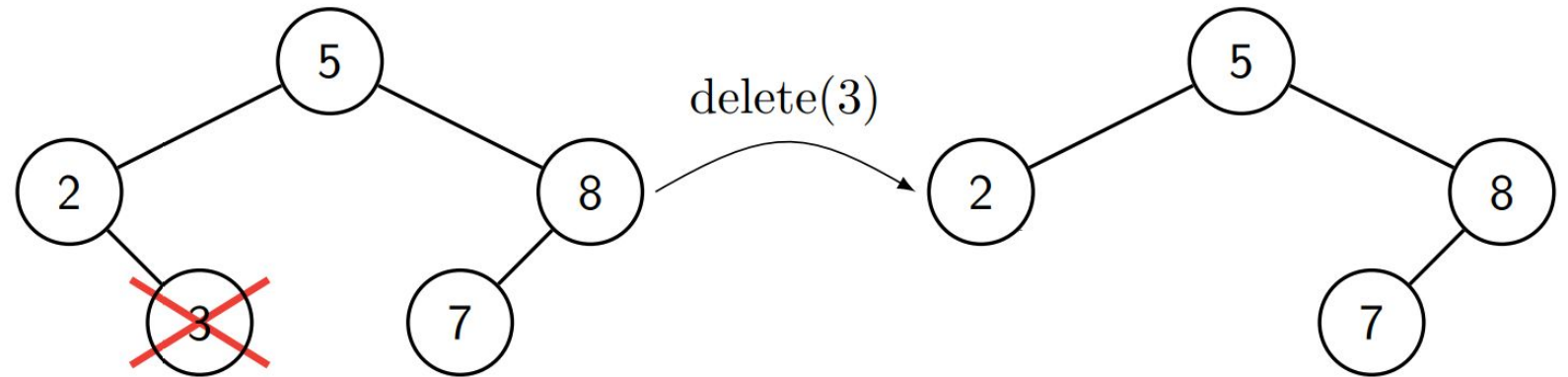
Entfernen

Das Entfernen (remove) eines Knotens erfordert das Suchen des Knotens und das Löschen dieses Knotens, so dass die Binärbaum-Struktur erhalten bleibt.

Dabei unterscheiden wir 3 Fälle:

1. Löschen eines Blatts
2. Löschen eines Knotens mit 1 Kind
3. Löschen eines Knotens mit 2 Kindern

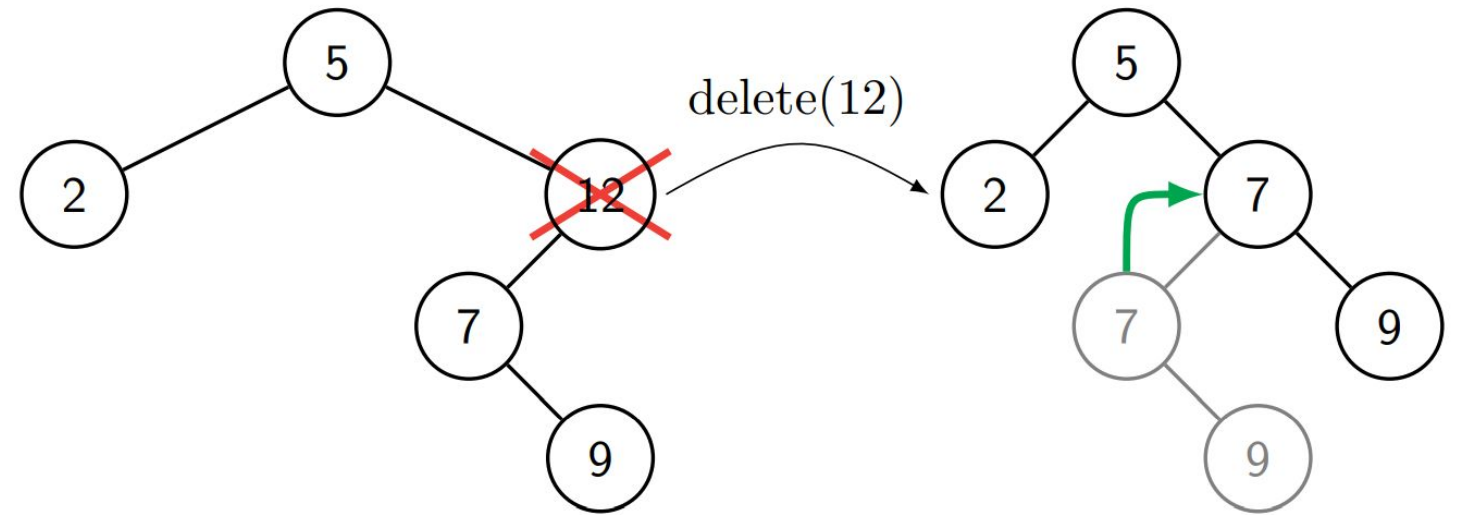
Binäre Suchbäume - Operationen



Entfernen - eines Blatts

1. Es wird nach dem zu entfernenden Knoten gesucht und dieser dann einfach entfernt.

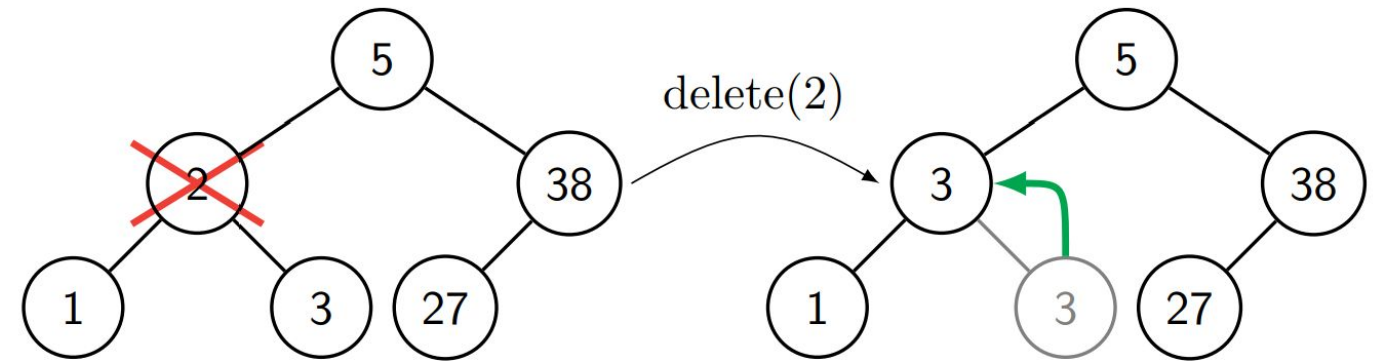
Binäre Suchbäume - Operationen



Entfernen - eines Knotens mit 1 Kind

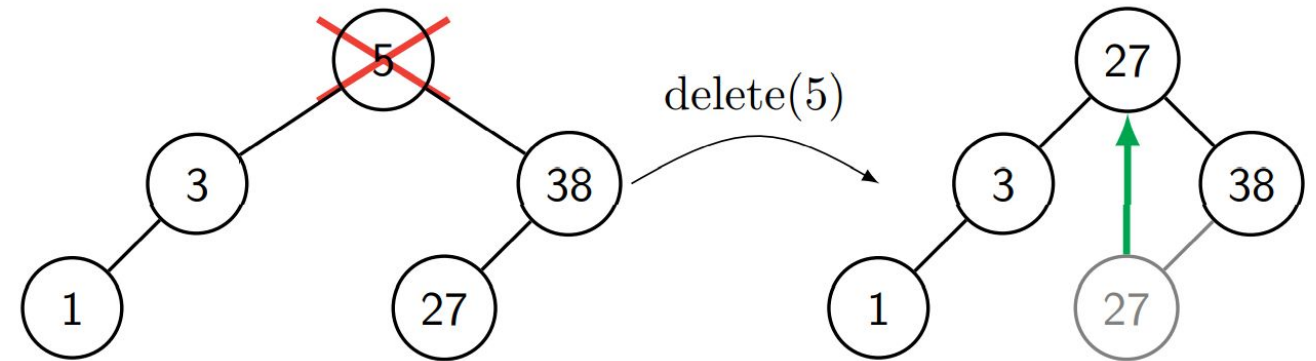
1. Es wird nach dem zu entfernenden Knoten gesucht. Dieser wird gelöscht und mit seinem Kind ersetzt.

Binäre Suchbäume - Operationen



Entfernen - eines Knotens mit 2 Kindern

1. Es wird der *symmetrische Nachfolger* bestimmt. Es ist derjenige Knoten, welcher im rechten Teilbaum des zu entfernenden Knotens am weitesten links steht.
2. Der Nachfolgeknoten wird von seinem alten Platz gelöscht (Fall 1 oder Fall 2) und ersetzt den zu löschenden Knoten.

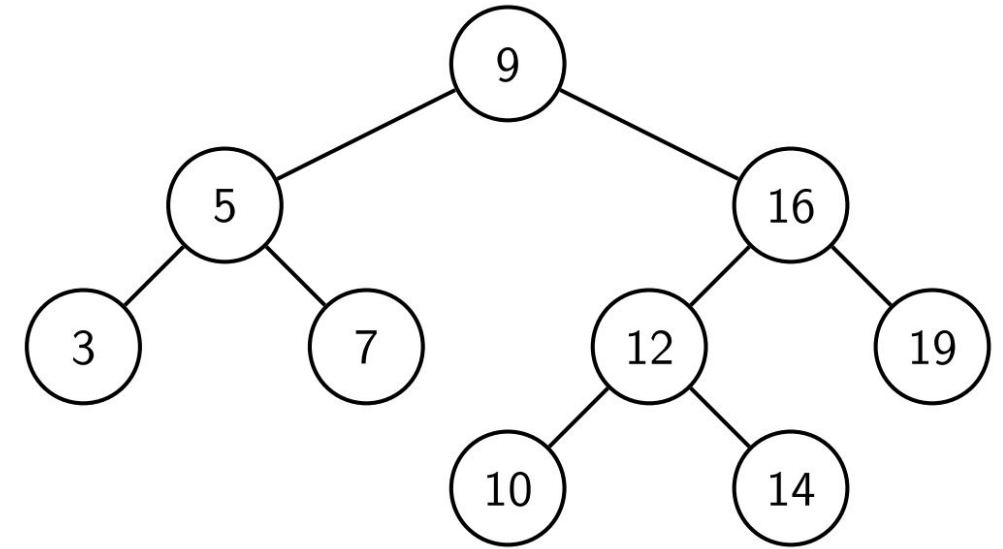


Binäre Suchbäume - Operationen

Aufgabe

Entfernen Sie im folgenden Baum die Knoten in der Reihenfolge: 5, 7, 16, 14, 9, 10.

Notieren Sie bei jedem Knoten, den Sie entfernen, um welchen Fall es sich handelt.

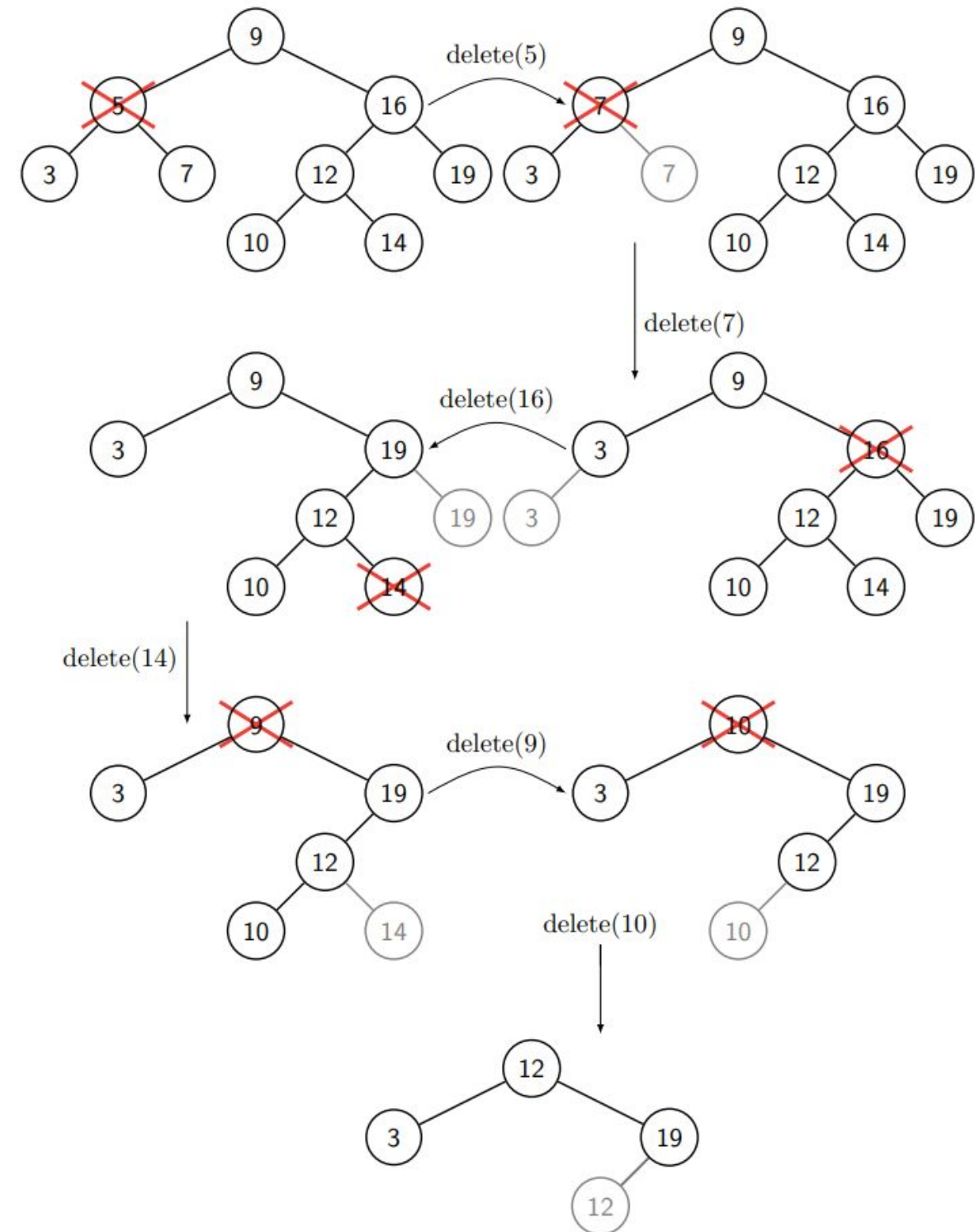


Binäre Suchbäume - Operationen

Aufgabe

Entfernen Sie im folgenden Baum die Knoten in der Reihenfolge: 5, 7, 16, 14, 9, 10.

Knoten	Fall
5	3
7	2
16	3
14	1
9	3
10	3



Binärbäume

Aufgabe:

Arbeitsblatt Aufgaben 9 - 12

Binäre Suchbäume - Programmieren

```
private static class Node<K extends Comparable<? super K>, E> {  
    final K key;  
    E element;  
    Node<K, E> left, right;  
  
    Node(K key, E element) {  
        this.key = key;  
        this.element = element;  
    }  
  
    public K getKey() { return key; }  
  
    public Node<K, E> getLeft() { return left; }  
  
    public Node<K, E> getRight() { return right; }  
}
```


Binäre Suchbäume - Programmieren

```
class BinarySearchTree<K extends Comparable<? super K>, E> implements Tree<K, E> {  
    private Node<K, E> root = null;  
    private int nodeCount = 0;  
  
    public Node<K, E> getRoot() { return root; }  
    public int size() { return nodeCount; }  
    public boolean isEmpty() { return root == null; }  
  
    public void insert(K key, E element) { root = insert(root, key, element); }  
    private Node<K, E> insert(Node<K, E> p, K key, E element) { TODO }  
  
    public E search(K key) { return search(root, key); }  
    private E search(Node<K, E> p, K key) { TODO }  
  
    public void remove(K key) { root = remove(root, key); }  
    private Node<K, E> remove(Node<K, E> node, K key){ TODO }
```

Binäre Suchbäume - Programmieren

```
class BinarySearchTree<K extends Comparable<? super  
    private Node<K, E> root = null;  
    private int nodeCount = 0;
```

```
    public Node<K, E> getRoot() { return root; }  
    public int size() { return nodeCount; }  
    public boolean isEmpty() { return root == null;
```

```
        public void insert(K key, E element) { root = insert(root, key, element); }  
        private Node<K, E> insert(Node<K, E> p, K key, E element) { TODO }
```

```
        public E search(K key) { return search(root, key); }  
        private E search(Node<K, E> p, K key) { TODO }
```

```
        public void remove(K key) { root = remove(root, key); }  
        private Node<K, E> remove(Node<K, E> node, K key){ TODO }
```

Rekursive Funktionen in einem Baum:

- Starte immer in der Wurzel des entsprechenden Teilbaumes.
 - Dies Wurzel muss als Argument übergeben werden.
- Gib den veränderten Teilbaum zurück an den Vater
 - Damit zeigt am Schluss der Operation root auf den aktualisierten Baum.

Bäume - Lernziele

Bäume allgemein

- Sie können die Konzepte von Bäumen benennen und erklären.
- Sie können Algorithmen zum Verarbeiten aller in einem Baum gespeicherten Elemente erklären und programmieren.

Binäre Suchbäume

- Sie können Algorithmen zum Einfügen, Suchen und Löschen in Binären Suchbäumen erklären und kennen wichtige Eigenschaften, wie Voraussetzungen und asymptotische Komplexität.
- Sie können diese grundsätzlichen Algorithmen auf binären Suchbäumen in Java implementieren.

Hausaufgaben

Arbeitsblatt alle noch nicht gelösten Aufgaben

Programmieren Aufgaben 1 & 2

Quellen:

Alle Zeichnungen von: [Binäre Suchbäume \(Leitprogramm\) – EducETH - ETH-Kompetenzzentrum für Lehren und Lernen | ETH Zürich](#)

(Dieses Leitprogramm habe ich Ihnen auch noch beigelegt als mögliches Skript.)