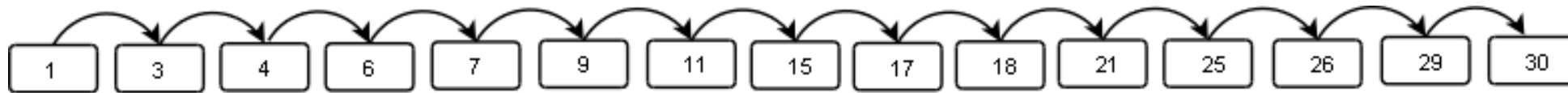


2. Die rekursive Datenstruktur Baum

2.1 Von der Liste zum Baum



In einer sortierten Liste ist das Suchen eines bestimmten Elements sehr einfach:

Sucht man beispielsweise das Element 9, wird die Liste durchlaufen und man erhält das Element nach 6 Vergleichen.

Sucht man das Element 10, sind 7 Vergleiche notwendig. Da $11 > 10$ gilt, kann 10 in der sortierten Liste nicht vorhanden sein.

Ist das gesuchte Element jedoch am Ende der Liste sind sehr viele Vergleichsoperationen notwendig, für das Element 30 benötigt man 15 Vergleiche. Bei großen Listen (z.B. ein Wörterbuch) hat eine solche Suche ein sehr schlechtes Zeitverhalten.

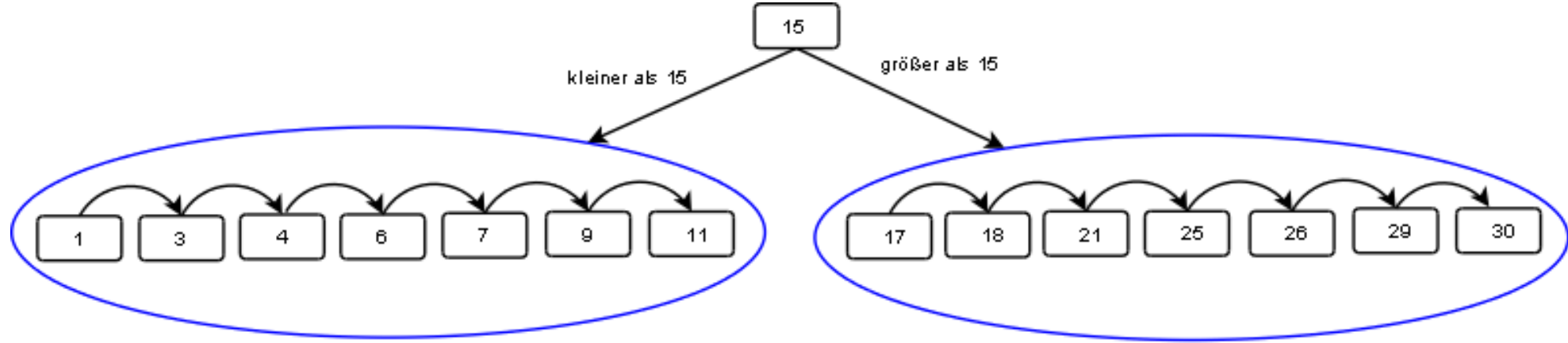
Aufgabe

Wie groß ist die durchschnittliche Anzahl von Vergleichen in der abgebildeten Liste?

Wie groß ist sie in einer geordneten Liste mit n Elementen?



Die Suche wird effizienter, wenn man die Liste in der Mitte teilt:



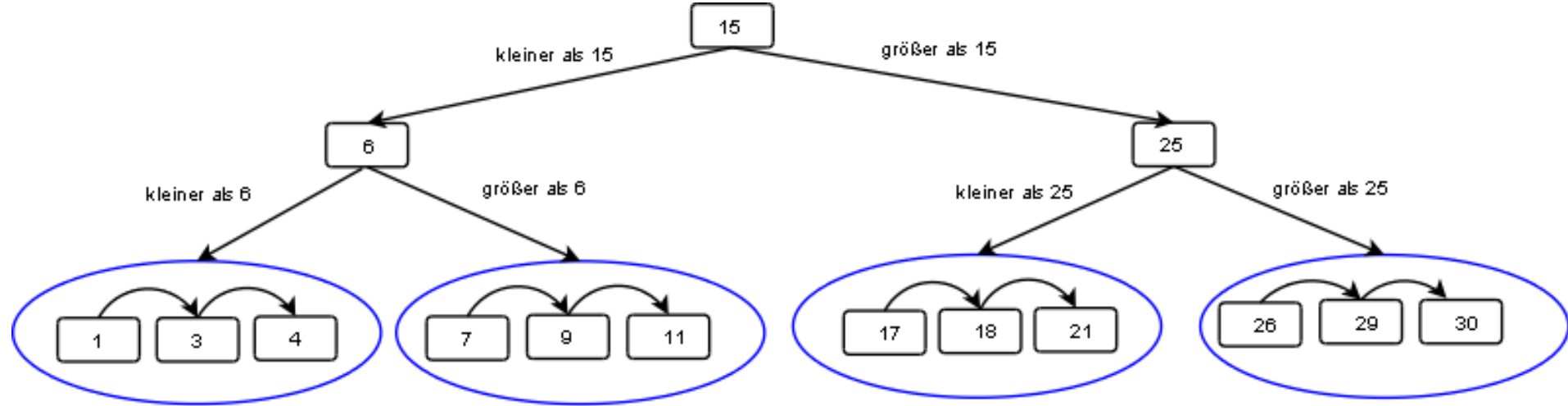
Für das Element 30 benötigt man nur mehr 8 Vergleiche.

Aufgabe

Wie groß ist die durchschnittliche Anzahl von Vergleichen?



Setzt man die Teilung für jede Liste fort, gelangt man zu folgender Struktur:



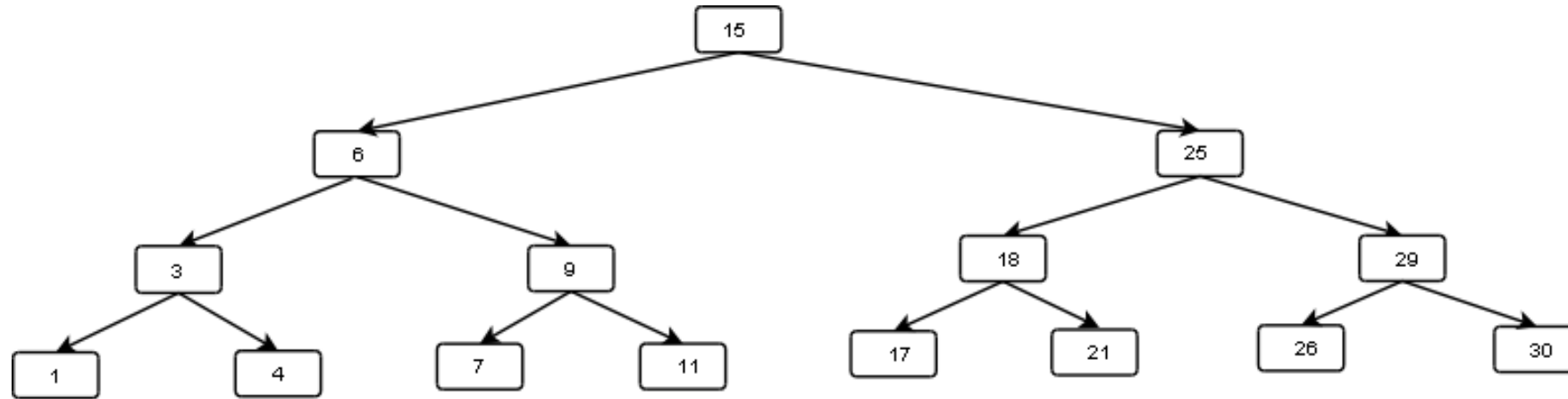
Aufgabe

Wie viele Vergleiche benötigt man für das Element 30?

Wie groß ist die durchschnittliche Anzahl von Vergleichen?



Nach der letzten Teilung erhält man:



Aufgabe

Wie viele Vergleiche benötigt man für das Element 30?

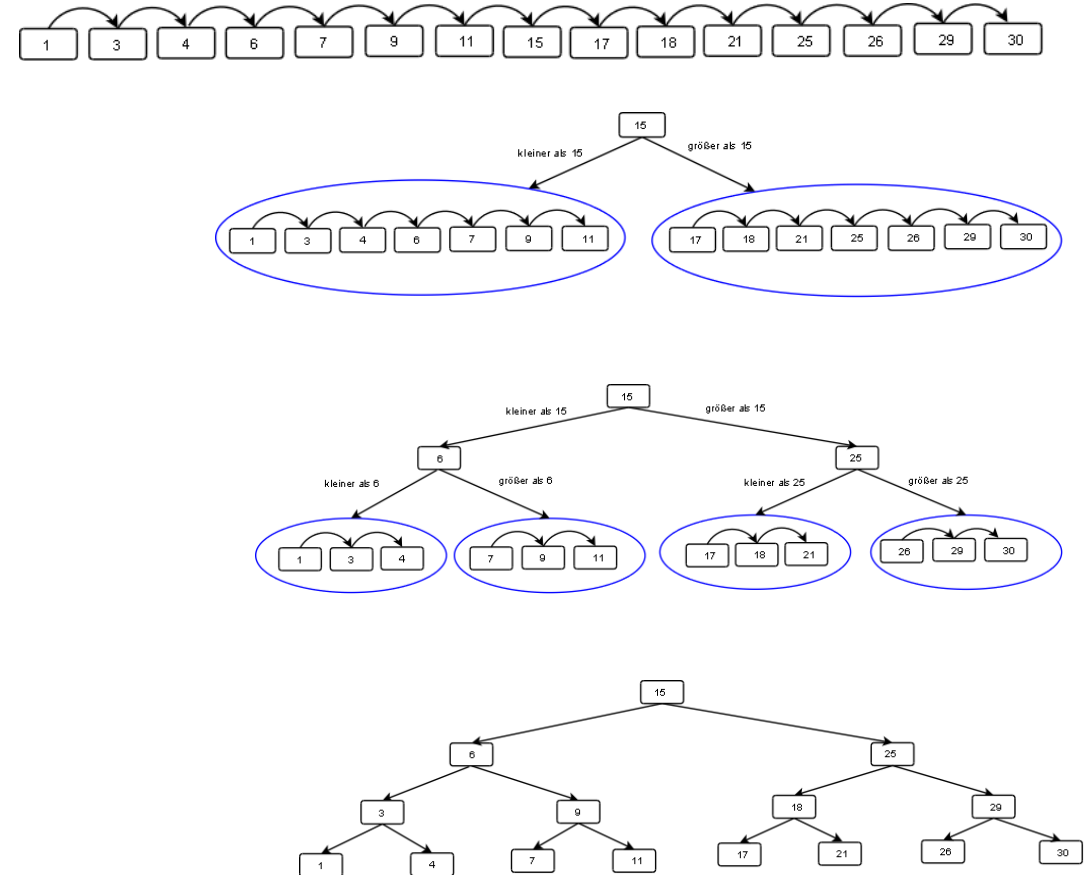
Wie groß ist die durchschnittliche Anzahl von Vergleichen?



Definition der Datenstruktur **Baum**

In einem Baum können die zugehörigen Objekte jeweils mehrere Nachfolger haben. Zusätzlich gilt:

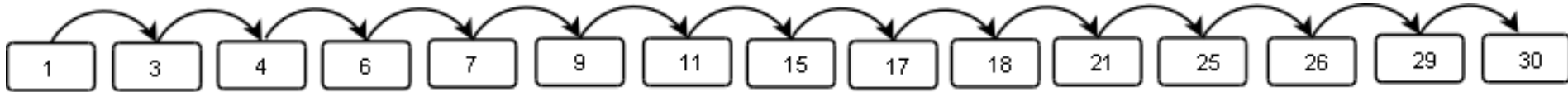
1. Genau ein Objekt wird nicht referenziert. Man nennt es Wurzel des Baumes
2. Alle anderen Objekte werden genau einmal referenziert. Man nennt sie Knoten. Jeder Knoten lässt sich von der Wurzel auf genau einem Weg erreichen.



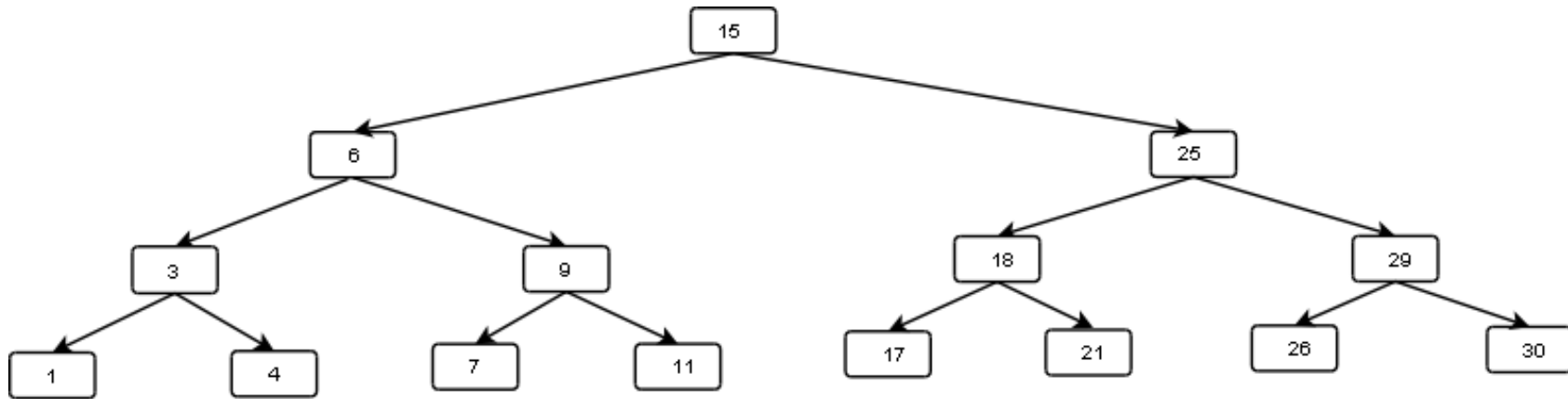


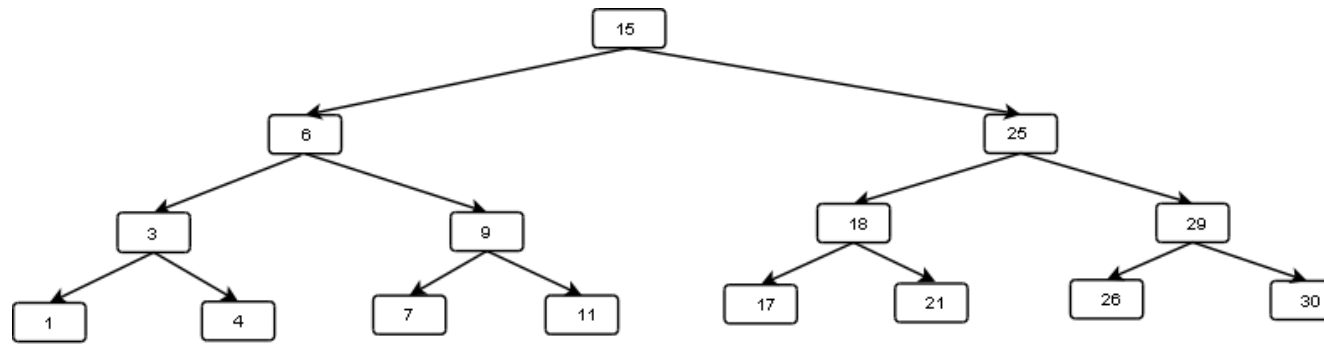
Überlege:

Ist eine Liste auch ein Baum?



Inwiefern ist ein Baum eine rekursive Datenstruktur?





Fachbegriffe

Hat jeder Knoten höchstens zwei Nachfolger, spricht man von einem **Binärbaum**.

Bei den Knoten unterscheidet man zwischen **inneren Knoten** (mit Nachfolger) und **Blättern** (kein Nachfolger).

Die Referenzen zwischen den Knoten nennt man **Kanten**.

Die **Tiefe eines Knotens** ist die Anzahl der Kanten, die beim Durchlauf von der Wurzel bis zum Knoten beschriftet werden.

Alle Knoten mit der gleichen Tiefe beschreiben eine **Ebene des Baumes**.

Die **Höhe des Baumes** ist festgelegt durch die größtmögliche Tiefe.

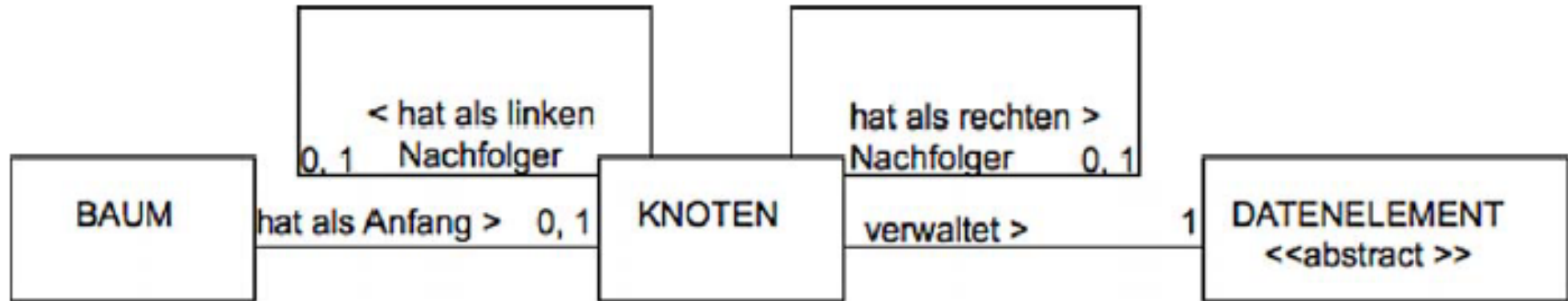
Anstelle der Begriffe Nachfolger-Vorgänger liest man in der Literatur auch oft Kind-Eltern, Sohn-Vater, Tochter-Mutter o.ä.

In einem **geordneten Baum** sind alle Elemente im linken Teilbaum kleiner (größer) und im rechten Teilbaum größer (kleiner) als die Wurzel. Dies gilt für alle Teilbäume.

Ein geordneter Binärbaum heißt auch **binärer Suchbaum**.

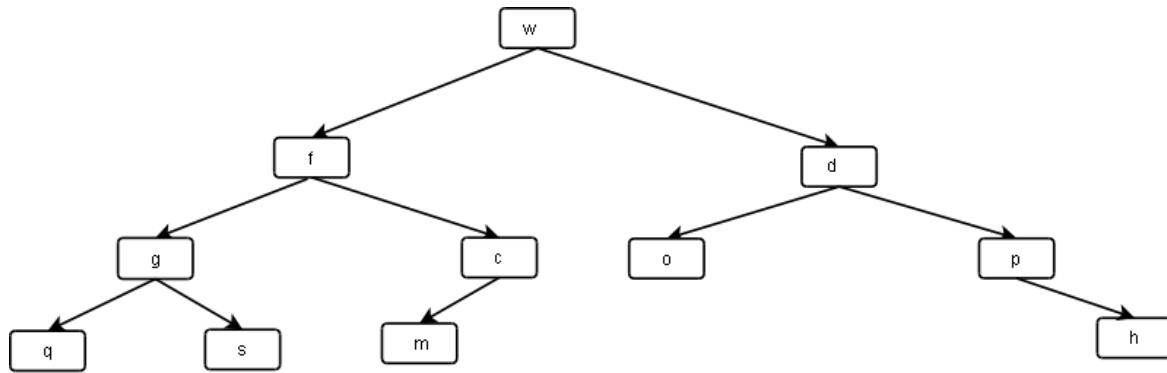


Klassendiagramm eines Binärbaumes:

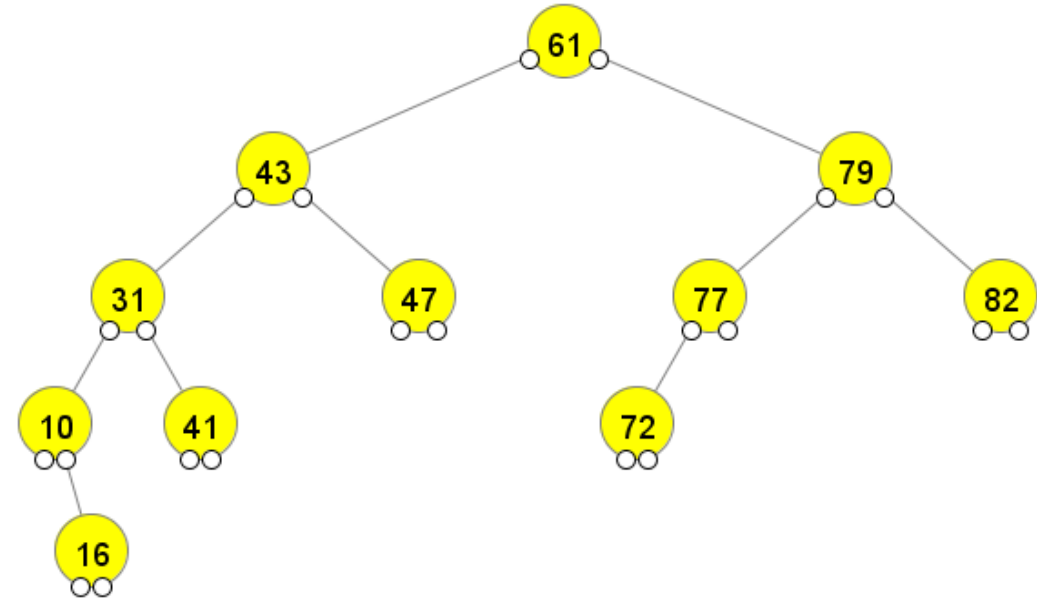




Ungeordneter Binärbaum:



Geordneter Binärbaum:



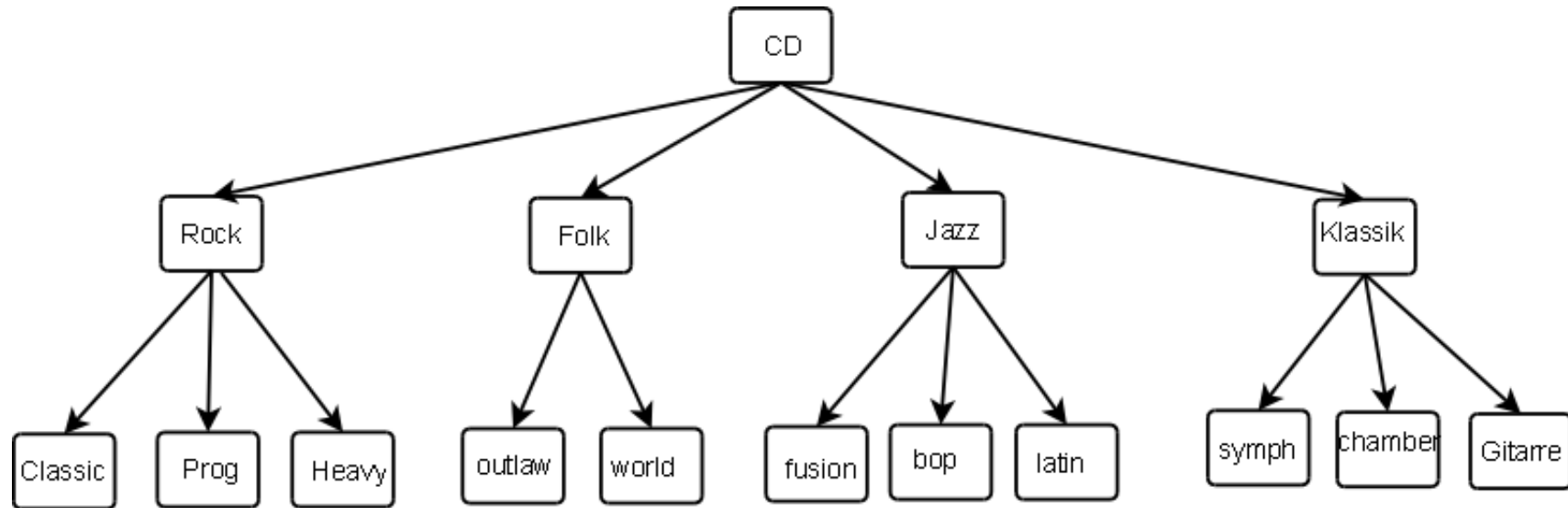


Klassendiagramm eines allgemeinen Baumes:





Beispiel:



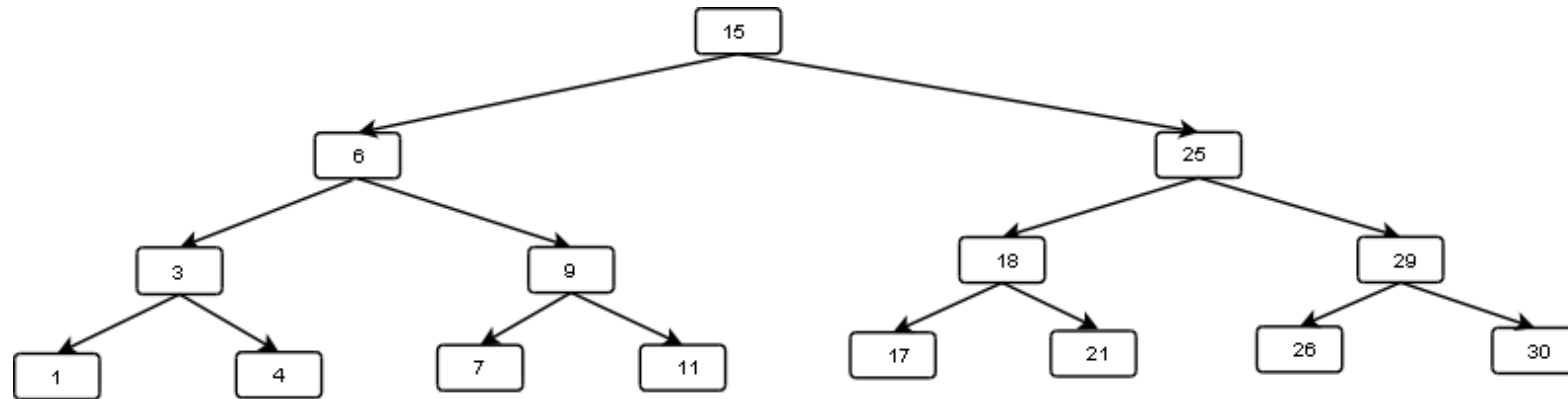


Aufgabe

Gib jeweils für einen Binärbaum mit $z = 1, 2, 3, 4, \dots$ Ebenen die Anzahl n der Knoten an.
Gehe davon aus, dass alle Ebenen die maximale Knotenzahl haben.

Wie lautet der allgemeine Zusammenhang zwischen der Anzahl z der Ebenen und der Anzahl n der Knoten?

Welche Näherung ist für große Werte von n möglich?





Übung 1



Experimentiere mit einem Baum-Visualisierungs-Tool:

<https://www.cs.usfca.edu/~galles/visualization/BST.html>

oder:

<http://www.bvt.ct-labs.de/>

Dort kann man geordnete Binärbäume erzeugen. Vollziehe an einigen Beispielen nach, wie der Baum aus einer geordneten Liste entstanden sein könnte. Beachte, dass die Listenteilung nicht immer in der Mitte erfolgen muss.

Bestimme auch die Tiefen der Knoten und die Baumhöhe.

Bearbeite im Buch die Aufgaben 4 und 6 auf den Seiten 68 – 69.



Übung 2



<http://www.matheprisma.de/Module/BinSuch/index.htm>

(Suchspiel und Suchbaum)