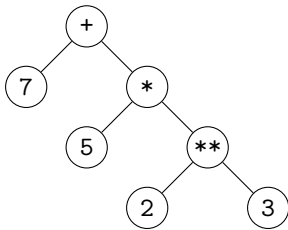
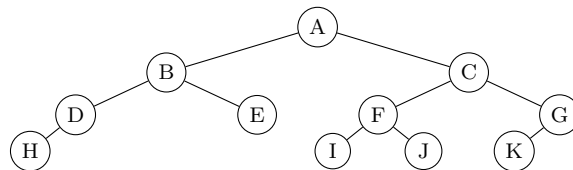


Aufgabe 1

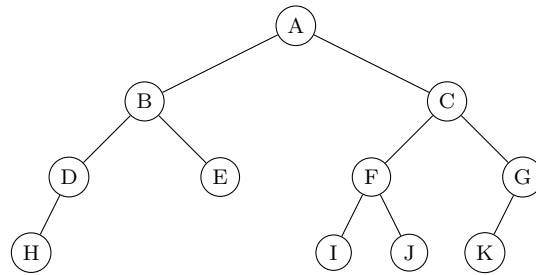
- HTML-Seiten
- hierarchische Dateisysteme
- Parse-Trees

Aufgabe 2 $7 + 5 * 2 ** 3$ **Aufgabe 3**

- Welche Kinder hat der Knoten mit dem Schlüssel C? F, G
- Welche Geschwister hat der Knoten H? *keine*
- Welche Eltern hat der Knoten mit dem Schlüssel E? B
- Welches sind die Blätter des Baums? H, E, I, J, K
- Welches sind die inneren Knoten des Baums? A, B, C, D, F, G
- Welche Tiefe hat der Knoten mit dem Schlüssel E? 2
- Welche Höhe hat der Knoten mit dem Schlüssel J? 0
- Welche Höhe hat der Baum? 3

Aufgabe 4

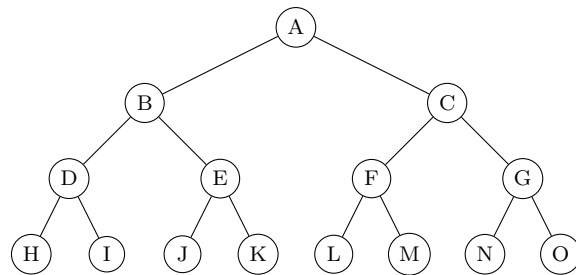
balancierter Binärbaum



Die Tiefe der Blätter unterscheiden sich höchstens um 1.

Aufgabe 5

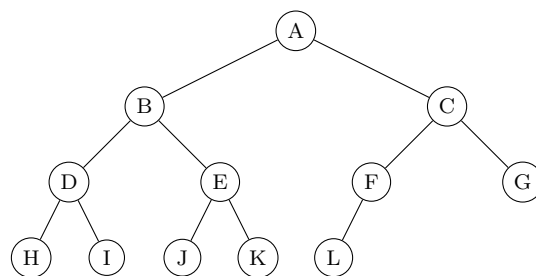
perfekter Binärbaum



Ein voller Binärbaum, dessen Blätter alle dieselbe Tiefe haben.

Aufgabe 6

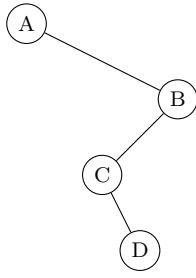
vollständiger Binärbaum



Ein balancierter Binärbaum, bei dem die tieferen Blätter alle links stehen.

Aufgabe 7

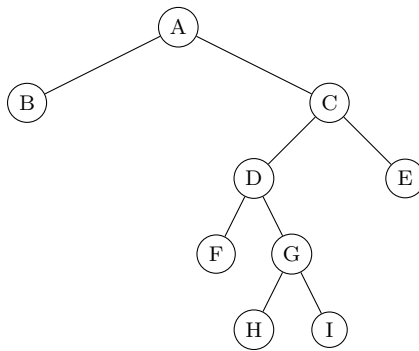
entarteteter Baum



Jeder Knoten hat höchstens ein Kind.

Aufgabe 8

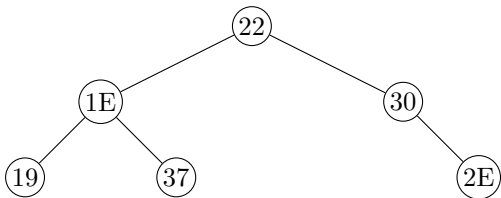
voller Binärbaum



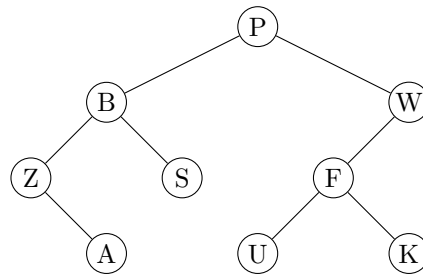
Alle inneren Knoten haben zwei Kinder.

Aufgabe 9

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	⊗	00	0B	41	1E	2F	10	00	00	00	00	00	00	00	00	22
1	37	00	00	00	3A	0F	00	32	41	00	00	00	00	00	00	00
2	00	00	00	00	00	32	00	00	00	0C	14	17	00	00	46	19
3	00	00	22	04	3A	00	00	32	00	00	30	00	4C	24	46	00
4	00	00	00	00	00	00	00	02	04	00	00	00	2E	00	00	00

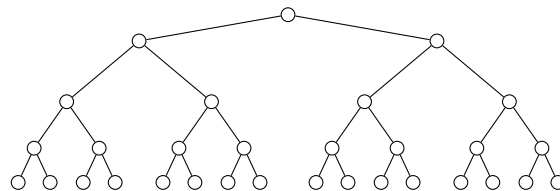


Aufgabe 10



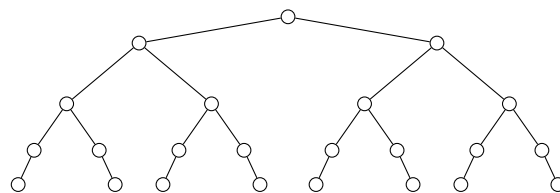
- (a) Preorder: P B Z A S W F U K
- (b) Inorder: Z A B S P U F K W
- (c) Postorder: A Z S B U K F W P

Aufgabe 11



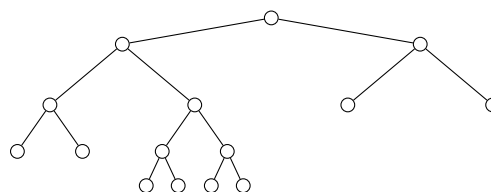
perfekter Binärbaum

Aufgabe 12



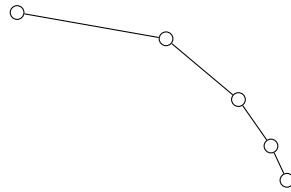
balancierter Binärbaum

Aufgabe 13



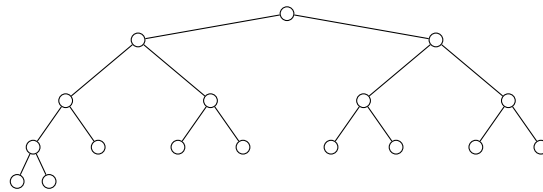
voller Binärbaum

Aufgabe 14



entarteter (degenerierter) Binärbaum

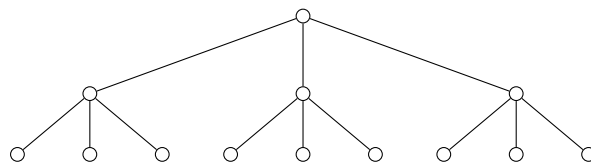
Aufgabe 15



balancierter *und* voller Binärbaum

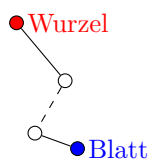
Aufgabe 15

perfekter ternärer Baum der Höhe 2



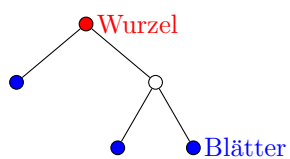
Aufgabe 16

- (a) Ein entarteter Baum hat genau ein Blatt. *wahr*



- (b) Ein voller Binärbaum hat eine gerade Anzahl Blätter. *falsch*

Gegenbeispiel mit 3 Blättern:

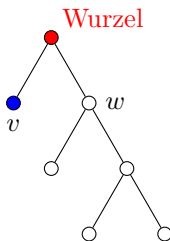


- (c) Wenn man in einem Baum eine Kante löscht, erhält man einen Wald. *wahr*

Beachte, dass ein isolierter Knoten auch ein Baum ist.

- (d) Wenn ein Baum die Höhe 3 hat und einer seiner Knoten v die Tiefe 1, dann muss v die Höhe 2 haben. *falsch*

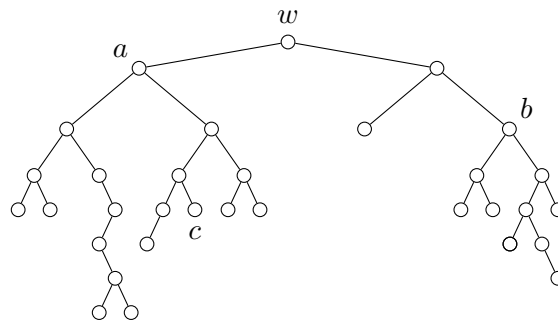
Gegenbeispiel: Obwohl es immer mindestens einen Knoten gibt, der diese Eigenschaft hat (w), kann es Knoten geben, für welche die Aussage falsch ist: Knoten v hat auch die Tiefe 1 aber die Höhe 0.



- (e) Ein perfekter Binärbaum der Höhe 3 hat 14 Kanten. *wahr*

Mach z. B. eine Zeichnung und zähle nach.

Aufgabe 17



Die *Tiefe eines Knotens* (depth, level) ist die Anzahl der Kanten auf dem Pfad vom Wurzelknoten zum betreffenden Knoten.

Die *Höhe eines Knotens* (height) ist die Anzahl der Kanten entlang des längsten Pfads vom Knoten zu einem Blatt, das im Teilbaum des Knotens liegt.

Die *Höhe eines Baums* (height of a tree) ist die Höhe des Wurzelknotens (Alternativ: Die maximale Tiefe eines Knotens.)

$$\text{Tiefe}(w) = 0, \text{Höhe}(w) = 7 = \text{Höhe}(\text{Baum})$$

$$\text{Tiefe}(a) = 1, \text{Höhe}(a) = 6$$

$$\text{Tiefe}(b) = 2, \text{Höhe}(b) = 4$$

$$\text{Tiefe}(c) = 4, \text{Höhe}(c) = 0$$

Aufgabe 18

Der Baum kann „algorithmisch“ gezeichnet werden:

Entferne die erste (öffnende) und letzte (schliessend) Klammer

Durchlaufe die Zeichenkette zeichenweise

wenn das Zeichen ein Wert ist:

schreibe ihn auf

wenn das Zeichen eine öffnende Klammer ist:

wenn links noch keine Kante ist:

gehe nach links

sonst:

gehe nach rechts

wenn das Zeichen eine schliessende Klammer ist:

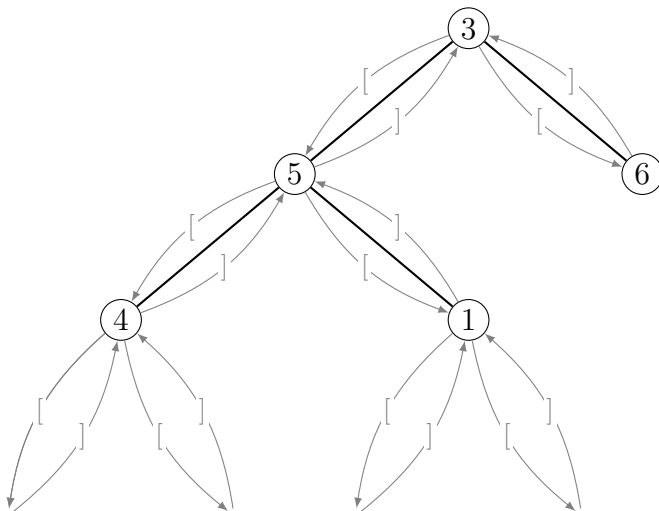
gehe entlang der Kante wieder zurück

wenn es ein anderes Zeichen (Komma, Leerzeichen) ist:

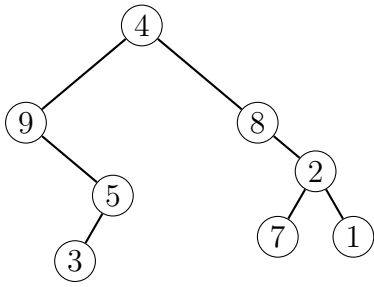
ignoriere es

Zeichne die Kanten ein, die beidseits einen Wert haben

$T = [3, [5, [4, [], []], [1, [], []]], [6, [], []]]$



Aufgabe 19



Die Aufgabe kann ähnlich wie die letzte gelöst werden:

- Beginne bei der Wurzel.
- Umfahre den Baum im Gegenuhrzeigersinn.
- Bei einem Knoten ist der Wert und ein Komma zu notieren.
- Für jede Bewegung nach unten schreibe [.
- Fehlt ein linker Knoten, schreibe [] und ein Komma.
- Fehlt ein rechter Knoten, schreibe [] .
- Für jede Bewegung nach oben schreibe] und ein Komma.
- Am Schluss setze [an den Anfang und] ans Ende der Zeichenkette.

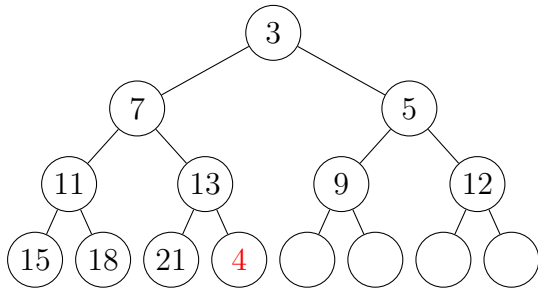
T = [4, [9, [] , [5, [3, [] , []] , []]] , [8, [] , [2, [7, [] , []] , [1, [] , []]]]]

Aufgabe 20

- als Vorrangwarteschlange (priority queue)
- als Datenstruktur für Heapsort

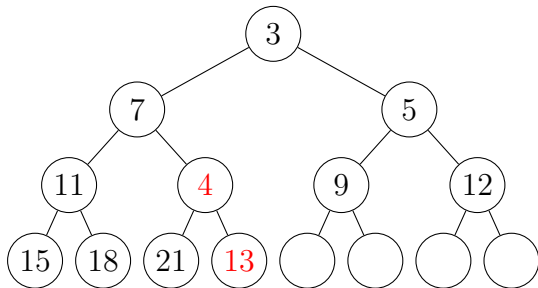
Aufgabe 21

Den neuen Wert 4 am Ende des Heaps einfügen.



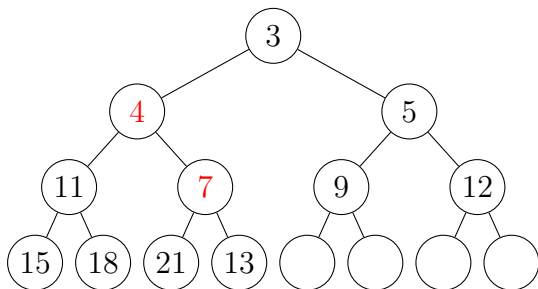
Der Knoten mit dem Wert 4 verletzt die Min-Heap-Eigenschaft.

Eine `swim()`-Operation mit 4 ausführen.



Der Knoten mit dem Wert 4 verletzt die Min-Heap-Eigenschaft.

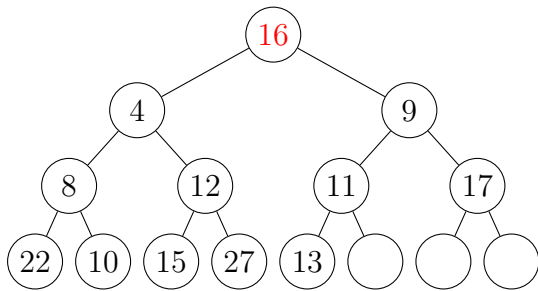
Eine weitere `swim()`-Operation mit 4 ausführen.



Die Min-Heap-Eigenschaft ist wieder hergestellt.

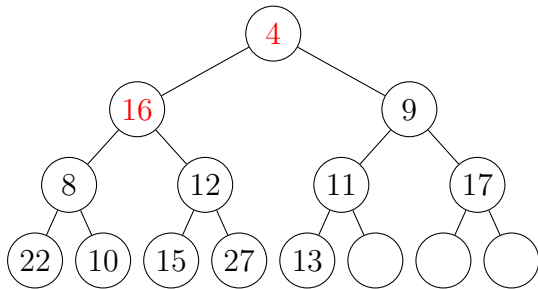
Aufgabe 22

Den letzten Wert (16) in den Wurzelknoten verschieben.



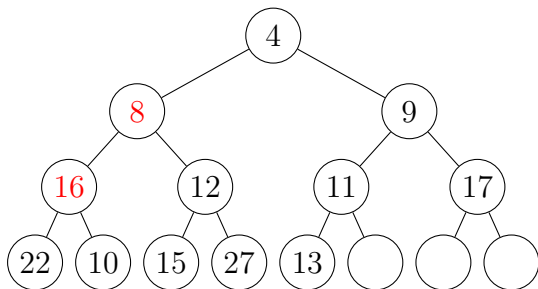
Der Knoten mit dem Wert 16 verletzt die Min-Heap-Eigenschaft.

Mit dem Wert 16 eine `sink()`-Operation durchführen.



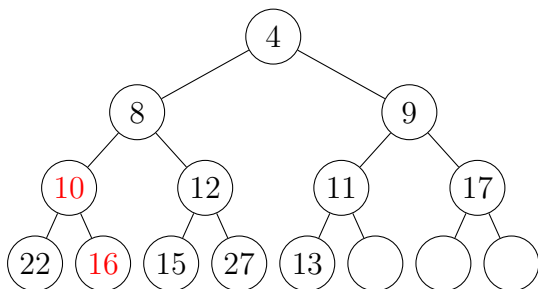
Der Knoten mit dem Wert 16 verletzt die Min-Heap-Eigenschaft.

Mit dem Wert 16 eine weitere `sink()`-Operation durchführen.



Der Knoten mit dem Wert 16 verletzt die Min-Heap-Eigenschaft.

Mit dem Wert 16 eine weitere `sink()`-Operation durchführen.



Die Min-Heap-Eigenschaft ist wieder hergestellt.

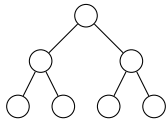
Aufgabe 23

Ein Heap besteht aus 68 Knoten.

- (a) Bestimme die Höhe h des Heaps.

Ein perfekter Baum der Höhe h hat $2^{h+1} - 1$ Knoten.

Beispiel: $h = 2$, $n = 2^3 - 1 = 7$ Knoten



Somit suchen wir $\min \{h \mid 68 \leq 2^{h+1} - 1\} \Rightarrow h = 6$

- (b) Ein Knoten hat den Listenindex 25. Welchen Index hat sein Elternknoten?

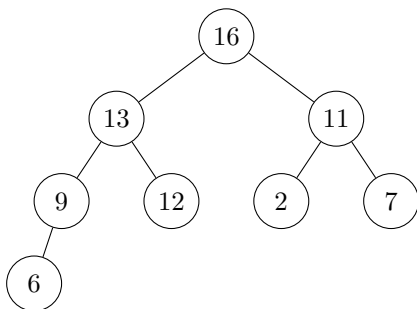
$$\lfloor 25/2 \rfloor = 12$$

- (c) Ein Knoten hat den Listenindex 34. Welchen Index hat sein rechtes Kind?

$$2 \cdot 34 + 1 = 69 \text{ [existiert nicht in einem Heap mit 68 Knoten]}$$

Aufgabe 24

- (a) $H = [\text{None}, 16, 13, 11, 9, 12, 2, 7, 6]$



- (b) Ja, da jeder Elternknoten mindestens so gross ist, wie alle seine direkten Kinder.