

Algorithmen und Datenstrukturen 1

Übungsblatt 4

Aufgabe 27.

Lösung.

	0	1	2	3	4	5	6	7	8	9	10	11	12
Lineares Sondieren			2			18	6	20	32	33	45		
Quadratisches Sondieren			2			18	6	20		32	33		45
Double Hashing	33		2			18	6	20			45	32	

Aufgabe 30.

Lösung.

- (a)
- (1) Nicht möglich, da 357 kleiner als 364 ist und die Suche an dieser Stelle nach links geht anstatt nach rechts.
 - (2) Nicht möglich, da 212 kleiner als 292 ist und sich im rechten Subtree von 292 befindet.
 - (3) Nicht möglich, da 898 links von 897 ist und damit die Voraussetzung, dass alle Elemente im linken Subtree kleiner sind als im Wurzelknoten verletzt worden ist.
 - (4) Möglich.

(b)

```

1 function validate(list [], target, min, max):
2   for i in list:
3     if i < min or i > max then
4       return 0
5     elif i > target then
6       max ← i
7     elif i < target then
8       min ← i
9     elif i == target then
10      return 1
11   else:
12     return 0
13 return 1

```

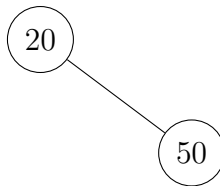
Aufgabe 31.

Lösung.

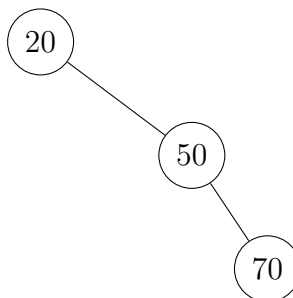
(a)



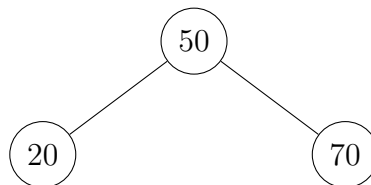
Starte mit 20 als einzigem Knoten.



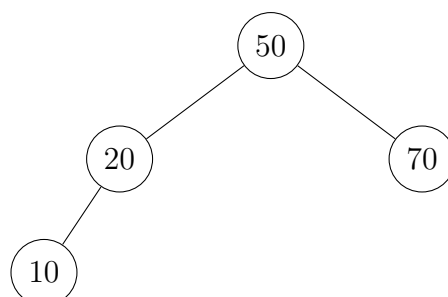
Füge 50 als rechten Knoten von 20 ein.



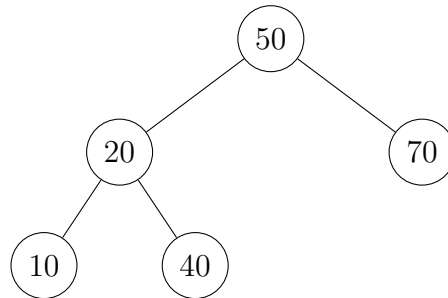
Füge 70 als rechten Knoten von 50 ein. Überprüfe die Balance → mehr als 2 Unterschied → rebalance.



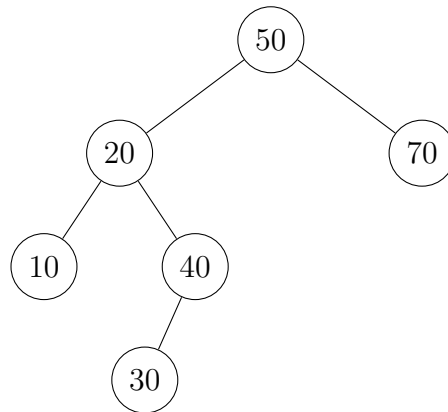
Rotiere den Graphen einmal nach links, sodass 50 in der Wurzel steht.



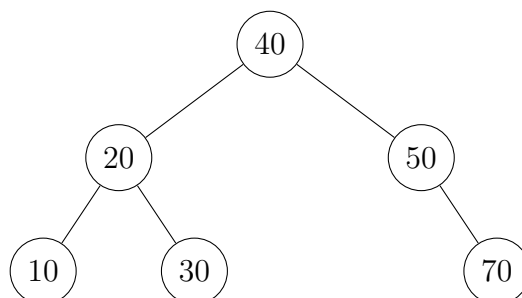
Füge 10 links von 20 ein, da kleiner als 50 und kleiner als 20. Kein balancing vonnöten, da Höhenunterschied nicht größer als 1.



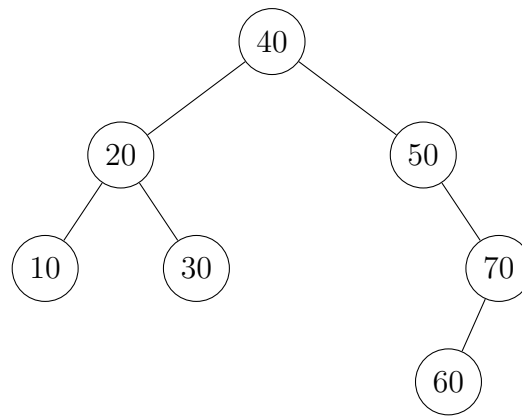
40 ist kleiner als 50, daher gehe in linken Subtree. Größer als 20, daher in rechten Subtree von 20. Noch kein Element vorhanden und an dieser Stelle einfügen. Balancing wieder keines notwendig, da Höhenunterschied nicht größer als 1.



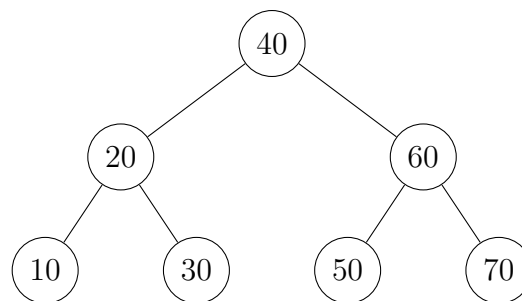
Füge 30 in linken Subtree von 40 ein, da kleiner als 50, größer als 20 und kleiner als 40. Höhenunterschied von linkem und rechtem Subtree von 50 ist größer als 1, daher balancing notwendig.



Hier wurde 40 zweimal nach rechts rotiert.

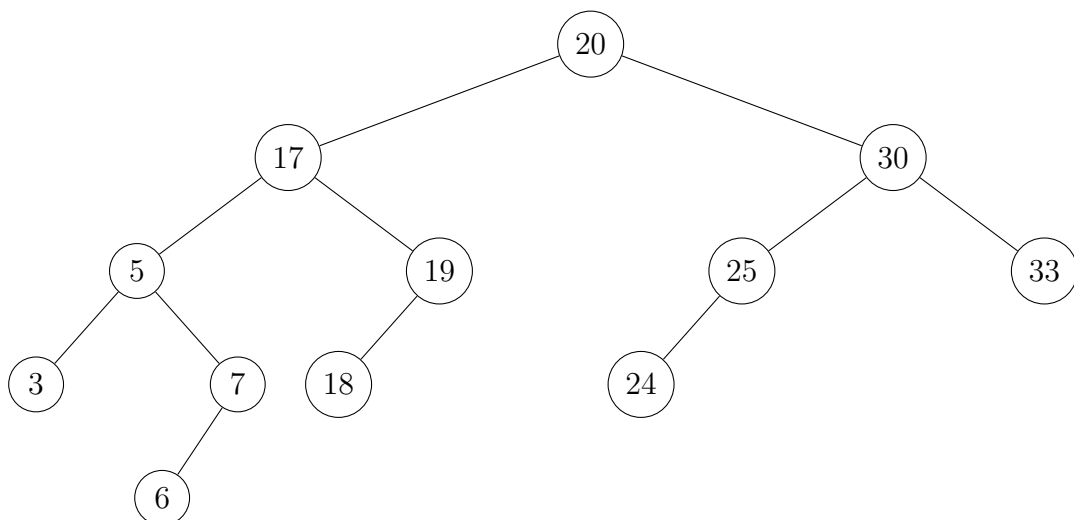


Höhenunterschied ist zwischen den Subtrees von 50 größer als 1, daher balancing notwendig.

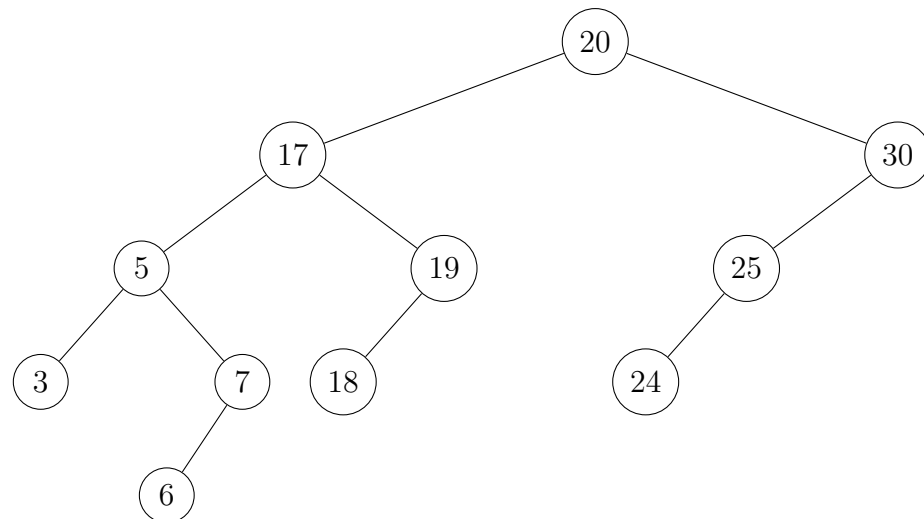


Endgültiger AVL-Baum.

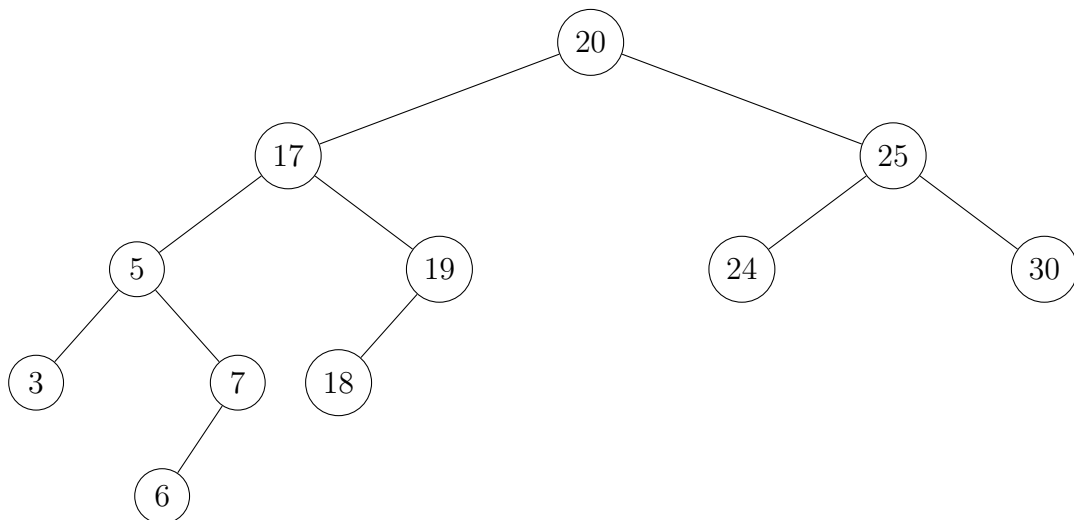
(b)



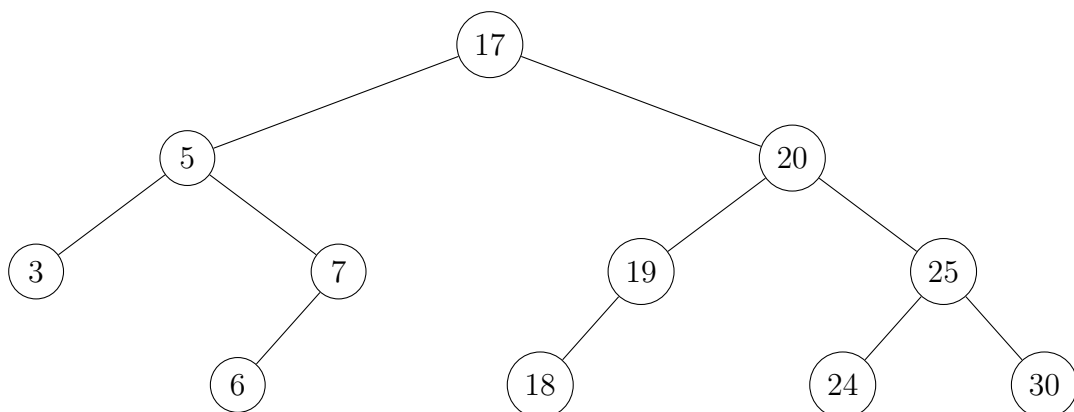
Nun 33 löschen.



Höhe von linkem Subtree von 30 ist um zwei größer als rechter Subtree von 30. → rotate right.



Linker Subtree ist um mehr als 1 größer als rechter Subtree.

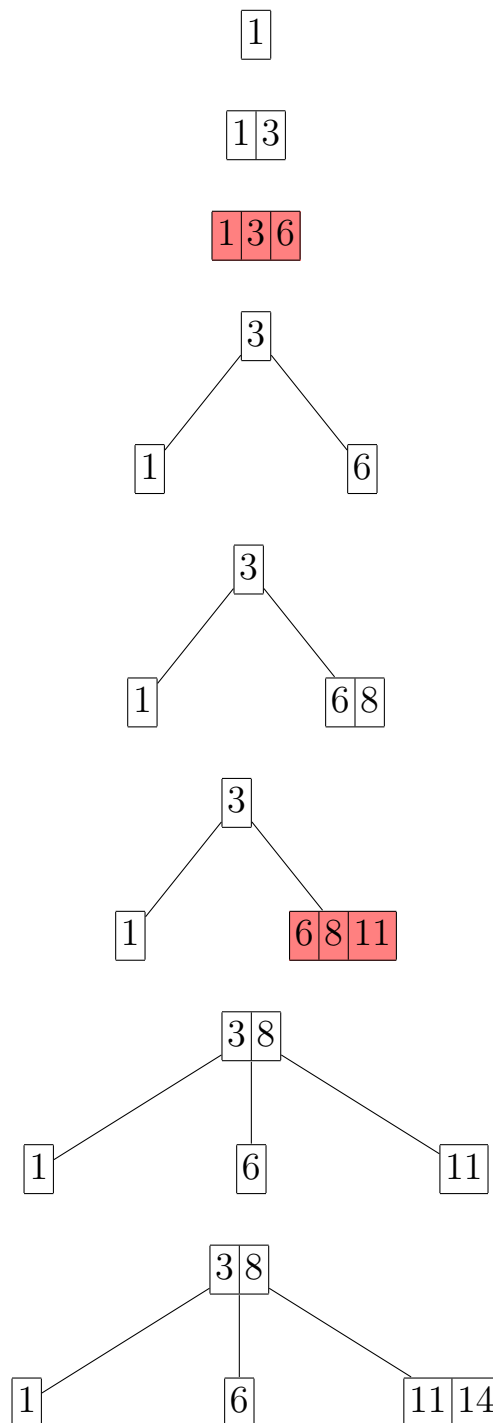


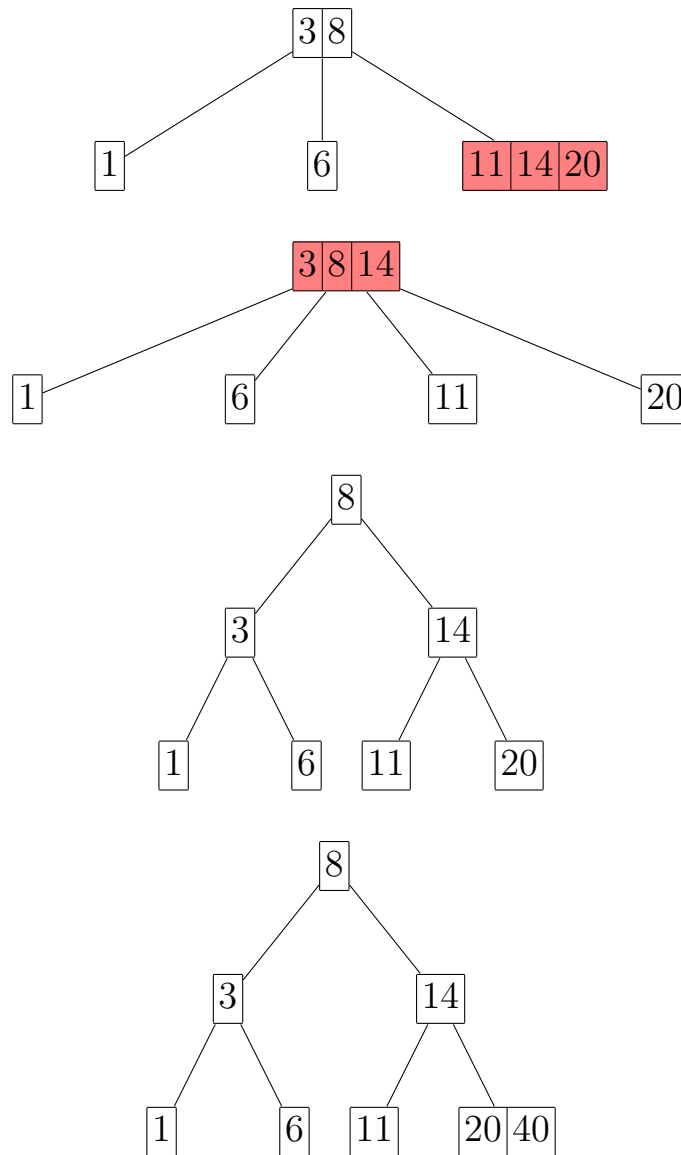
Endgültiger AVL-Baum.

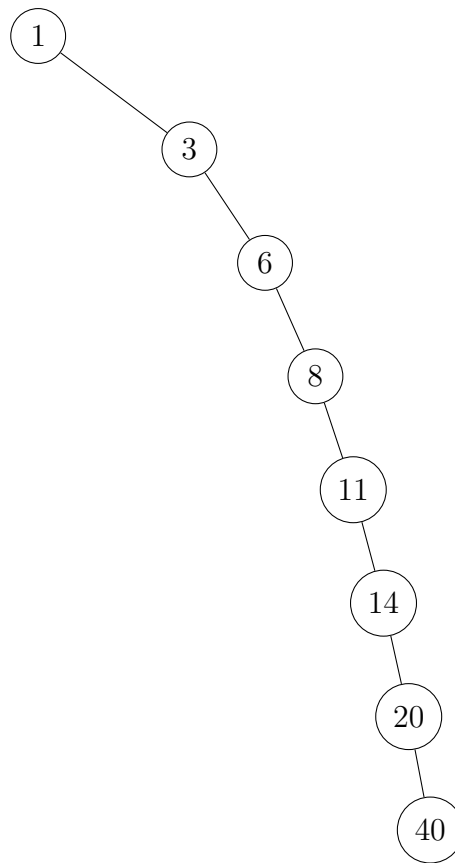
Aufgabe 32.

Lösung.

(a)







(b)

