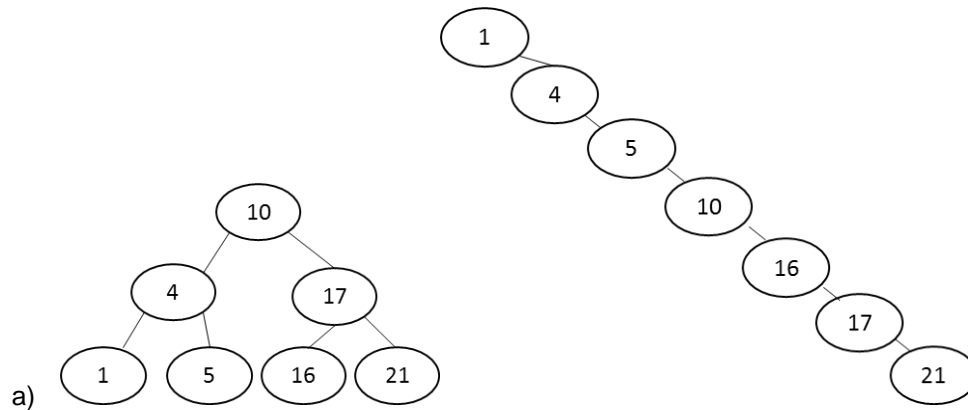




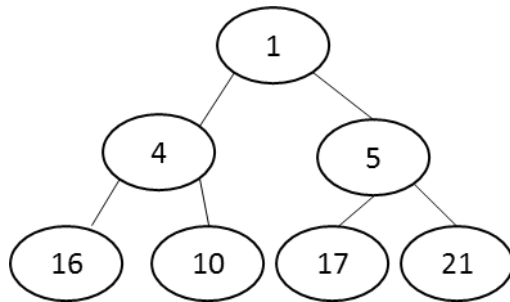
Lösung 07: Binäre Suchbäume

Aufgabe 1: Binäre Suchbäume

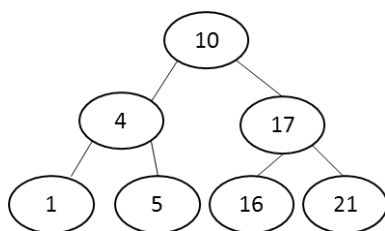


Hinweis: Für den rechten Fall gibt es auch eine andere Möglichkeit.

b) Es gibt mehrere mögliche MinHeaps, ein Beispiel siehe Skizze!



c) Achtung: Diese Aufgabe bezieht sich NICHT auf den MinHeap, sondern auf den binären Suchbaum!



Inorder: 1-4-5-10-16-17-21

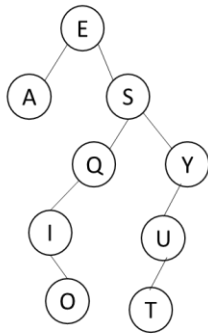
Preorder: 10-4-1-5-17-16-21

Postorder: 1-5-4-16-21-17-10

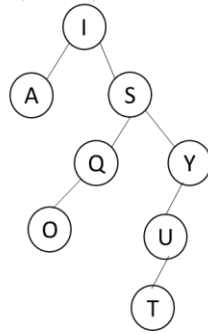
d) Siehe e)

e)

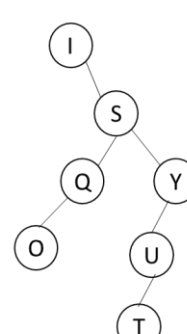
Nach dem Einfügen
aller Schlüssel



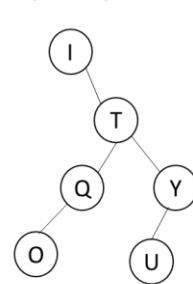
Löschen von E
(Fall 3b)



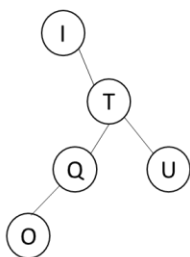
Löschen von A
(Fall 1)



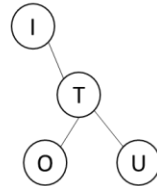
Löschen von S
(Fall 3b)



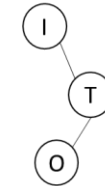
Löschen von Y
(Fall 2b)



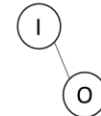
Löschen von Q
(Fall 2b)



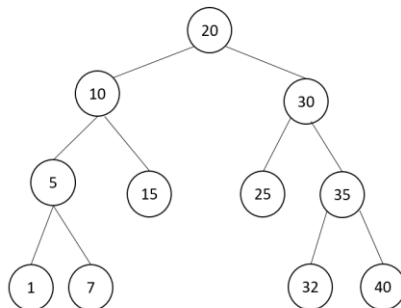
Löschen von U
(Fall 1)



Löschen von T
(Fall 3b)

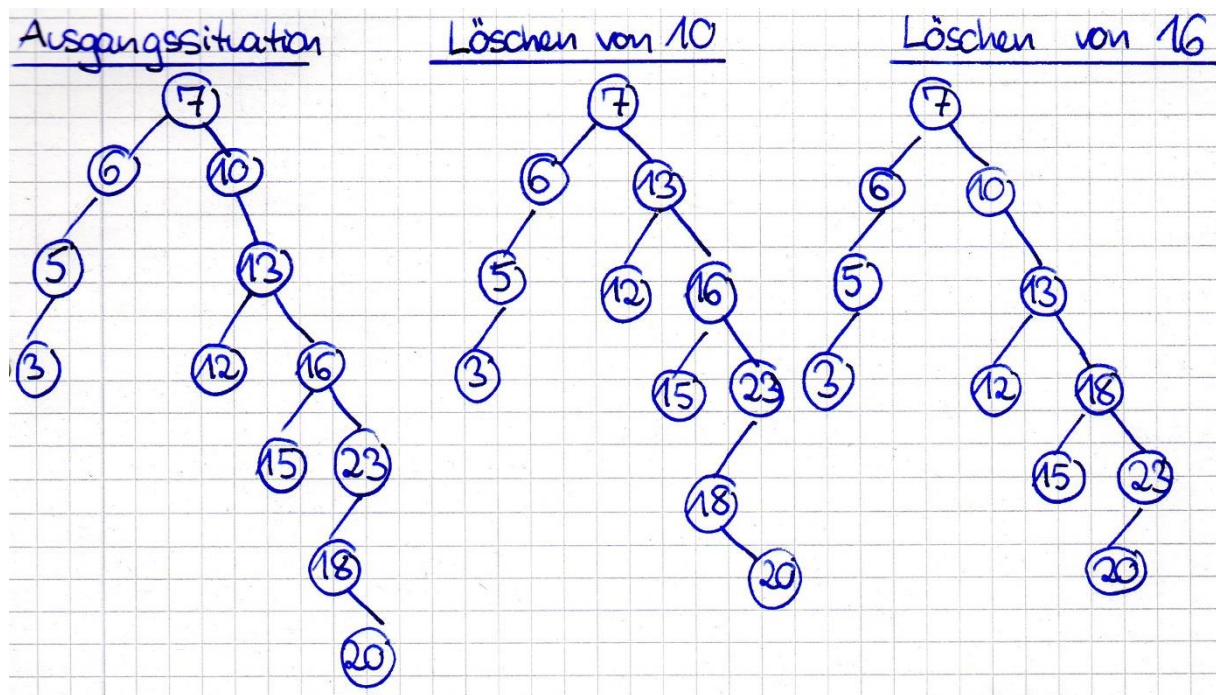


- f) Das erste Element muss die Wurzel sein (hier 20).
 Das nächste Element (10) kann das linke oder rechte Kind der Wurzel sein. Da 10 kleiner als 20 ist, muss es das linke Kind sein.
 Die 5 ist kleiner als die Wurzel 20 und kleiner als 10, muss also linkes Kind der 10 sein. usw.



Aufgabe 3: Löschen und Höhe

- a) Siehe Musterlösung im GitLab: https://inf-git.fh-rosenheim.de/muwo522/ad_wise_2019/tree/master/src/de/th_rosenheim/ad/uebung07/BST.java
- b) Siehe Skizze
- c) Siehe Skizze



- d) Siehe Musterlösung im GitLab: https://inf-git.fh-rosenheim.de/muwo522/ad_wise_2019/tree/master/src/de/th_rosenheim/ad/uebung07/BST.java