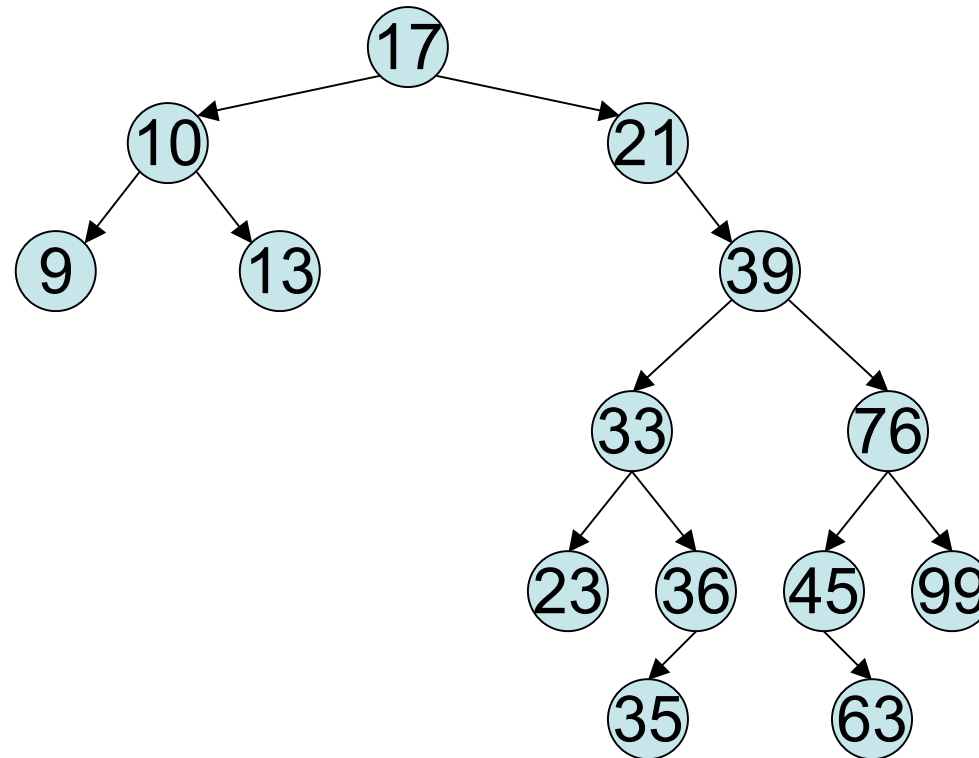


# Übung - Bäume



## **Aufgabe 2:**

Löschen Sie aus dem gegebenen binären Suchbaum nacheinander folgende Elemente: 13, 21, 39, 17. Wenden Sie die in der Vorlesung vorgestellten Strategien an. Geben Sie nach jedem Löschvorgang den resultierenden Baum an. Soll ein Knoten mit 2 Söhnen gelöscht werden, soll das Ergebnis beider Varianten angegeben werden ( "Löschen durch Verschmelzen" und "Löschen durch Kopieren").

# Übung - Bäume

## **Aufgabe 1:**

Schreiben Sie je eine Funktion, die ...

- (i) die Anzahl Blätter
- (ii) die Anzahl linker Söhne
- (iii) die Höhe

... eines binären Baumes bestimmt.

## **Aufgabe 3:**

- Schreiben Sie eine iterative Funktion zur Inorder-Durchmusterung eines binären Baumes. Jeder Knoten beinhaltet einen Integer Wert.
- Hinweis: Nutzen Sie die Datenstruktur STACK zur Zwischenspeicherung. Die Operationen push und pop müssen Sie nicht implementieren. Sie können diese als gegeben voraussetzen.

## **Aufgabe 4:**

Schreiben Sie eine Funktion, die einen binären Baum in Levelorder formatiert ausgibt (d.h. für jede neue Baumebene eine neue Zeile). Jeder Knoten enthält einen ganzzahligen Wert.

Hinweis: Nutzen Sie die Datenstruktur QUEUE. Diese ist als ADT (Klasse) „treeQueue“ mit folgenden Methoden gegeben:

*bool isEmpty()* -> Gibt „true“ zurück, falls QUEUE leer ist, andernfalls „false“

*void enqueue(n)* -> fügt das Element n in die QUEUE ein

*dequeue()* -> gibt das nächste Element aus der Queue aus