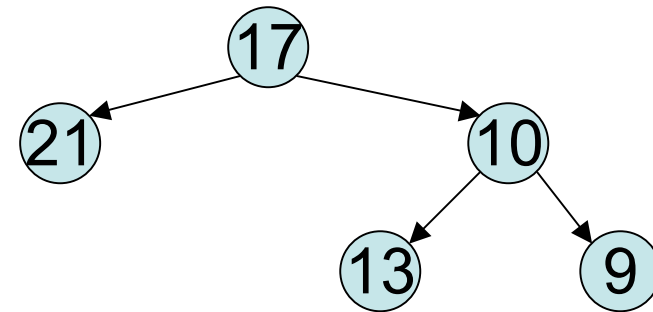
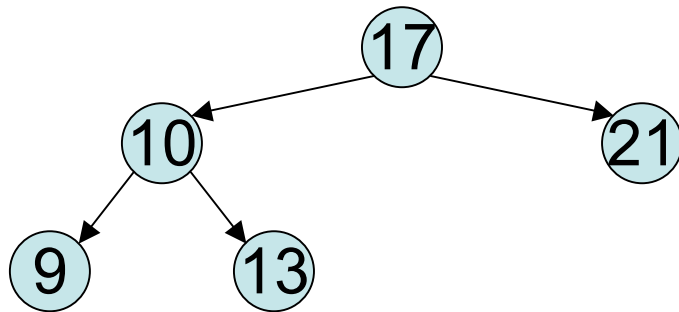


# Übung - Bäume

## Aufgabe 1:

Schreiben Sie eine Funktion, die einen gegebenen binären Suchbaum wie folgt spiegelt:



# Übung - Bäume

## Aufgabe 2:

Die Knoten eines binären Suchbaums wurden zusätzlich mit einer Balanceinformation versehen.

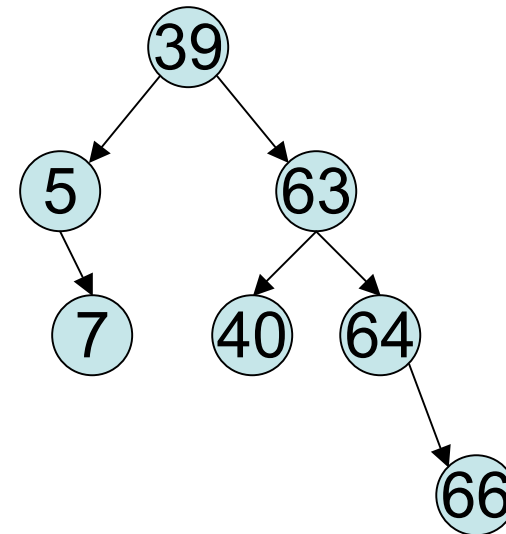
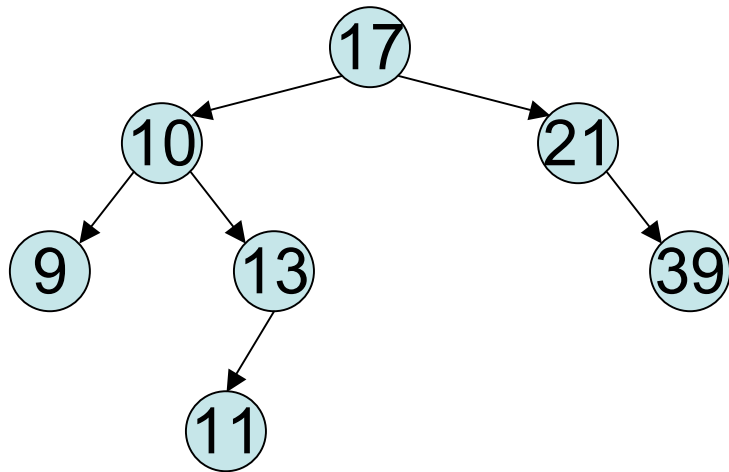
```
Struct node {  
    int info;  
    int balance;  
    node *left, *right;  
};
```

Die Balance eines Knotens entspricht der Differenz  
(Höhe des rechten Teilbaums) - (Höhe des Linken Teilbaums).

Modifizieren Sie die Einfügefunktion für binäre Suchbäume so, dass nach dem Einfügen alle Balanceinformationen wieder korrekt sind.

Hinweis: Vor dem Einfügen sind alle Balanceinformationen korrekt.  
Aktualisieren Sie deshalb **nur** die Knoten, deren Balance sich tatsächlich geändert hat.

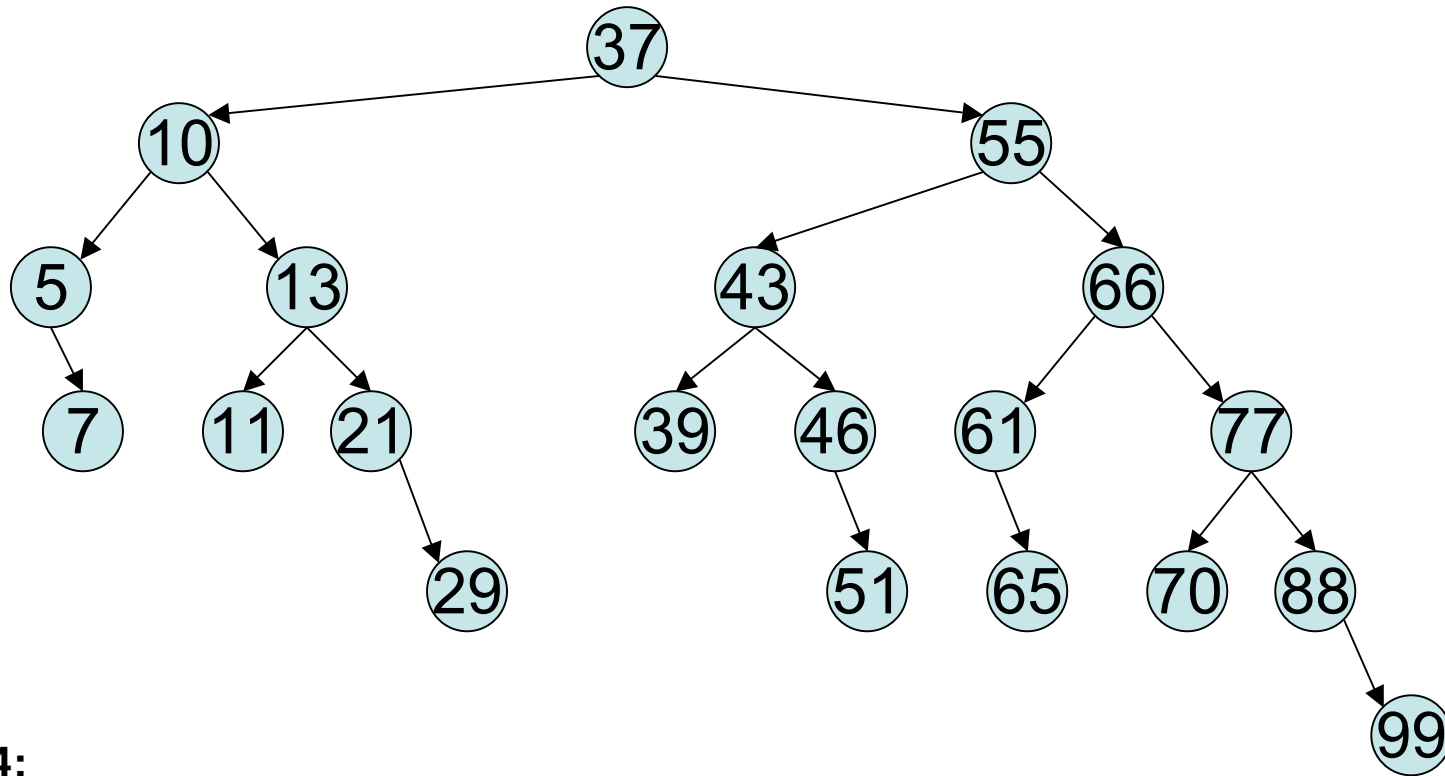
# Übung - Bäume



## **Aufgabe 3:**

Zeichnen Sie die vorliegenden AVL-Bäume nach dem Löschen des Elements 39. Geben Sie im Fall einer erforderlichen Ausbalancierung die jeweils benutzte Methode (Einfach- oder Doppelrotation) an.

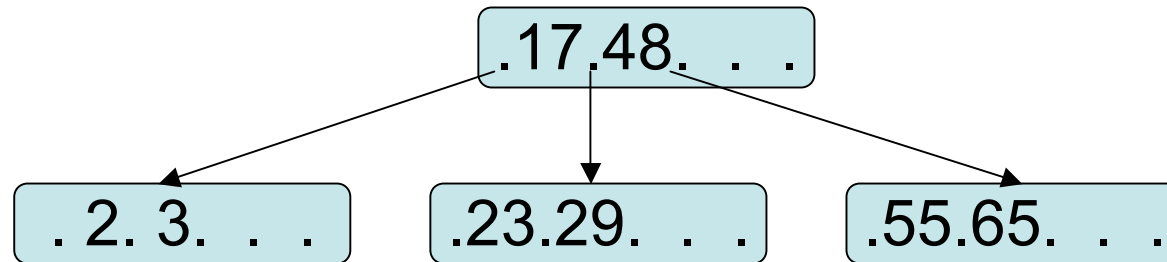
# Übung - Bäume



## **Aufgabe 4:**

Zeichnen Sie den vorliegenden AVL-Baum nach dem Löschen des Elements 7.  
Geben Sie im Fall einer erforderlichen Ausbalancierung des Baumes die jeweils benutzten Methoden (Einfach- oder Doppelrotation) an.

# Übung - Bäume



## **Aufgabe 5:**

Zeichnen Sie den vorliegenden B-Baum nach dem Löschen des Elements 48 und 2.