

# Algorithmen und Programmierung

## Übungsblatt 10

### WS 2022/23

Dr. Felix Jonathan Boes

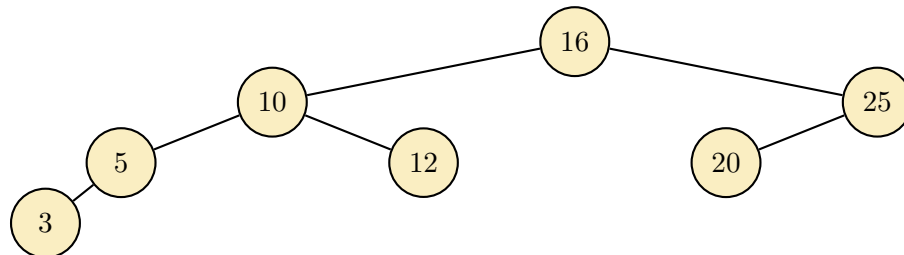
Benedikt Bastin, Ellen Bundschuh, Anna Höpfner, Gina Muuss, Adrian Oeyen, Felix Roth,  
Thore Wolf

Ausgabe: 12.12.2022

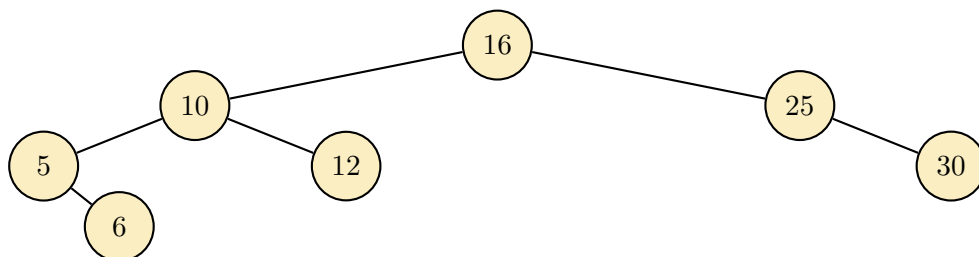
Abgabe: keine Abgabe; Präsentation in der Übung

**Aufgabe 1** (AVL-Bäume am Beispiel). Lösen Sie die folgenden Teilaufgaben mit Stift und Papier.

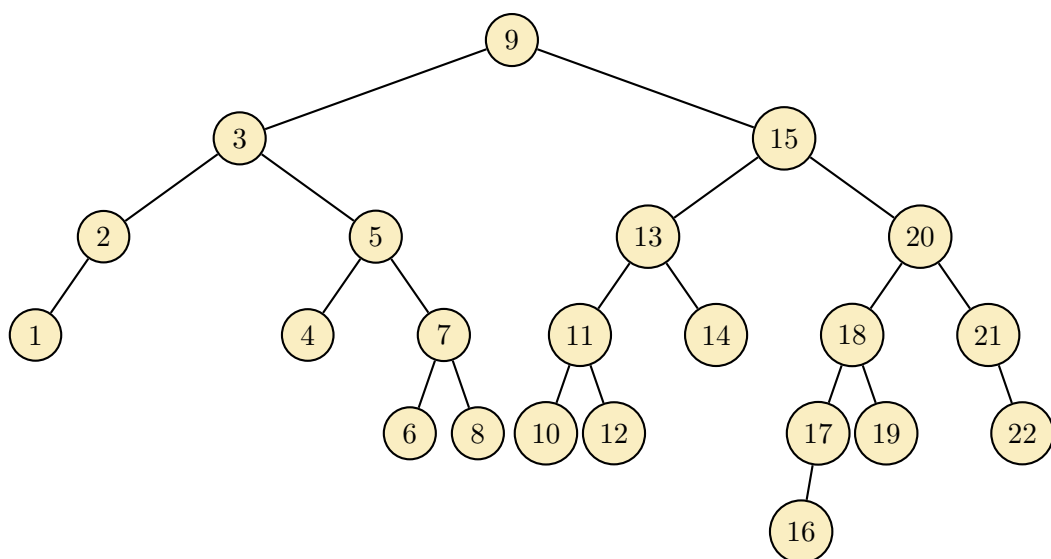
- (1) Betrachten Sie den folgenden AVL-Baum. Fügen Sie diesem AVL-Baum das Element 4 hinzu. Demonstrieren Sie beim Einfügen alle, in der Vorlesung diskutierten, Teilschritte und notieren Sie neben jeden Knoten den aktuellen Balancekoeffizienten.



- (2) Betrachten Sie den folgenden AVL-Baum. Entfernen Sie aus diesem AVL-Baum das Element 25. Demonstrieren Sie beim Entfernen alle, in der Vorlesung diskutierten, Teilschritte und notieren Sie neben jeden Knoten den aktuellen Balancekoeffizienten.



- (3) Betrachten Sie den folgenden AVL-Baum. Entfernen Sie aus diesem AVL-Baum das Element 2. Demonstrieren Sie beim Entfernen alle, in der Vorlesung diskutierten, Teilschritte.



**Aufgabe 2** (Binäre Suchbäume). In Ihrem Abgabe-Repository hat Flavius einen Branch mit dem Namen `Zettel_10` angelegt. Implementieren Sie die dort, unter Verwendung der Projektstruktur, die Klasse `BinarySearchtree` und demonstrieren Sie Ihre Implementierung. Sie dürfen natürlich Bestandteile Ihrer Implementierung der Klasse `BinaryTree` wiederverwenden.

Um das Projekt zu kompilieren, soll (im Stammverzeichnis des Projekts) folgender Compileraufruf verwendet werden. Alternativ verwenden Sie das Buildsystem `cmake`.

```
clang++ -std=c++17 -I./include -I./external -fsanitize=address \
  IHRE_QUELLDATEIEN \
  examples/aufg1.cpp -o aufg1
```

**Aufgabe 3** (Hashtable Teil 1). In Ihrem Abgabe-Repository hat Flavius einen Branch mit dem Namen `Zettel_10` angelegt. Implementieren Sie die dort, unter Verwendung der Projektstruktur, die Klasse `Hashtable` und demonstrieren Sie Ihre Implementierung. Sie dürfen Ihre Listenimplementierung verwenden. Dabei sollen Sie in dieser Woche Hashtables mit einer festen Bucketzahl von 32 und einer festgelegten Hashfunktion implementieren.

Um das Projekt zu kompilieren, soll (im Stammverzeichnis des Projekts) folgender Compileraufruf verwendet werden. Alternativ verwenden Sie das Buildsystem `cmake`.

```
clang++ -std=c++17 -I./include -I./external -fsanitize=address \
  IHRE_QUELLDATEIEN \
  examples/aufg2.cpp -o aufg2
```