

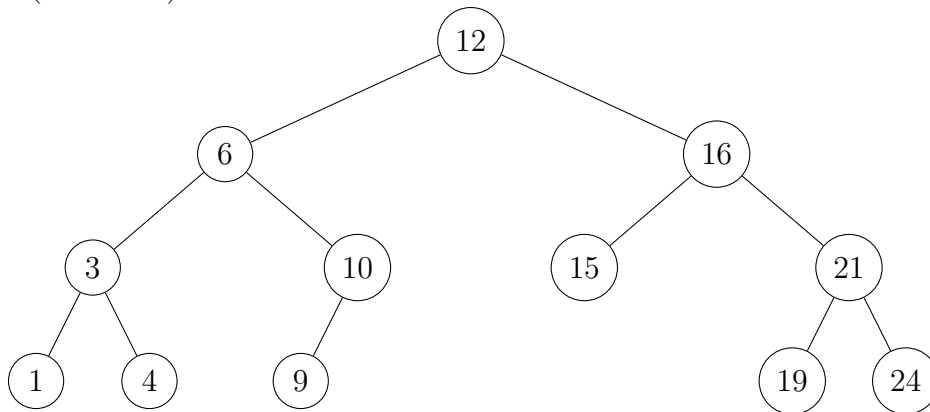
Petra Mutzel
Nicola Beume, Christian Bockermann, Christian Horoba,
Ingo Schulz, Dirk Sudholt, Christine Zarges

Sommersemester 2009

DAP2 Übung – Blatt 7

Ausgabe: 28. Mai, Abgabe: 4. Juni, 14:00 Uhr, Block: C

Aufgabe 7.1 (4 Punkte)



- Gib die Knoten des gegebenen binären Baums beim Durchlauf in Preorder-, Inorder- und Postorder-Reihenfolge aus. Ist der Baum ein binärer Suchbaum?
- Gegeben sind die folgenden Traversierungsreihenfolgen eines binären Baums. Rekonstruiere den Baum aus den Traversierungen und erkläre dein Vorgehen.

Preorder: 4, 2, 9, 8, 1, 6, 5, 3

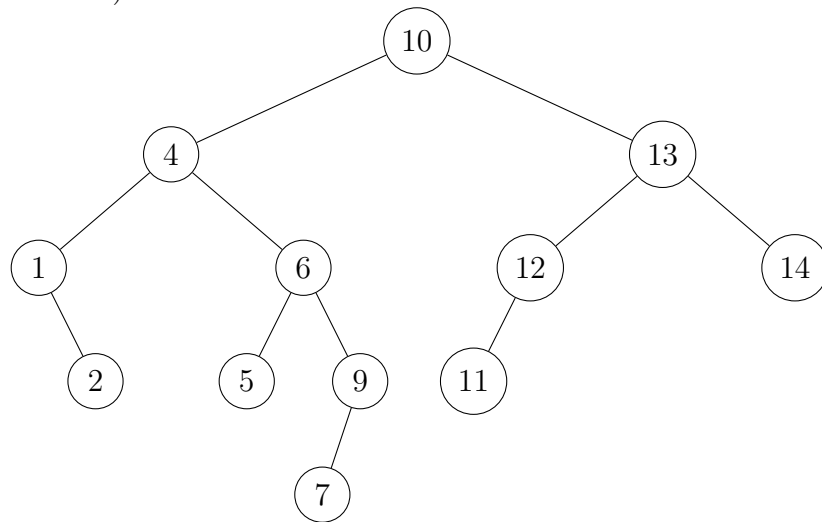
Inorder: 9, 2, 8, 4, 6, 1, 5, 3

Wie ändert sich das Vorgehen, wenn Postorder- und Inorder-Reihenfolge gegeben sind?

Aufgabe 7.2 (4 Punkte)

Gib jeweils für die Höhen 1, 2, 3 und 4 einen AVL-Baum an, der bei gegebener Höhe eine *minimale Anzahl* von Knoten enthält. Notiere zu allen Knoten den Balancegrad.

Aufgabe 7.3 (4 Punkte)



- Füge in den oben angegebenen AVL-Baum nacheinander Knoten mit den Schlüsseln 3 und 8 in dieser Reihenfolge ein. Zeichne den AVL-Baum nach jeder Rotationsoperation (eine Doppelrotation zählen wir als eine Rotation). Notiere jeweils die Art der ausgeführten Rotation.
- Betrachte wieder den oben angegebenen AVL-Baum *ohne* die in a) eingefügten Schlüssel. Lösche aus dem AVL-Baum den Knoten mit dem Schlüssel 10. Beschreibe das Vorgehen von DELETE und zeichne den Baum nach der Löschoption.
Achtung: Verwende dabei die Implementierung von DELETE aus Listing 4.6 im Skript, die die Funktion PREDECESSOR aufruft.
- Betrachte noch einmal den originalen AVL-Baum *ohne* die Operationen aus a) und b). Lösche aus dem AVL-Baum den Knoten mit dem Schlüssel 14. Zeichne den Baum nach jeder Rotationsoperation.

Präsenzaufgabe 7.4

Eine Inorder-Traversierung eines Suchbaums gibt die Elemente in aufsteigend sortierter Reihenfolge aus. Betrachte folgenden Sortieralgorithmus für ein Array A mit paarweise verschiedenen Einträgen.

```
1: procedure TREESORT( $A$ )
2:   for  $i := 1 \dots n$  do
3:     Füge  $A[i]$  in einen natürlichen binären Suchbaum ein.
4:   end for
5:   Gib die Inorder-Traversierung des Suchbaums aus.
6: end procedure
```

Gib jeweils obere Schranken für die Worst-Case- und die Best-Case-Laufzeit des Algorithmus in O -Notation an. Ist der Sortieralgorithmus in situ? Ist er adaptiv?

Was ändert sich, wenn anstelle eines natürlichen binären Suchbaums ein AVL-Baum verwendet wird?

In welchen Anwendungen hat das Sortieren mit Suchbäumen Vorteile gegenüber Sortieralgorithmen wie z. B. MergeSort?