

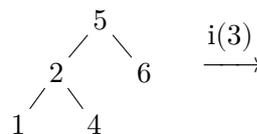
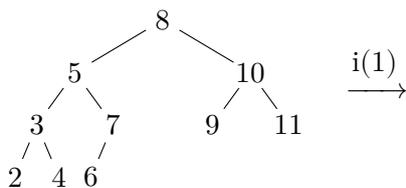
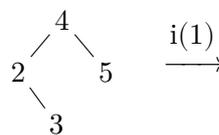
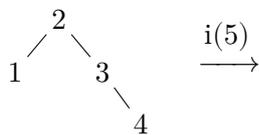
# Algorithmen und Datenstrukturen

## Aufgabe 1 (AGS 8.12)

Fügen Sie in die folgenden AVL-Bäume den jeweils angegebenen Schlüssel ein und stellen Sie die AVL-Eigenschaft her. Dokumentieren Sie dabei die ausgeführten Operationen unter Nutzung der folgenden Abkürzungen:

- $i(x)$  – Einfügen des Knotens mit Schlüsselwert  $x$
- $L(x)$  – Linksrotation um den Knoten mit Schlüsselwert  $x$
- $R(x)$  – Rechtsrotation um den Knoten mit Schlüsselwert  $x$

Geben Sie unmittelbar nach jedem  $i(x)$ -Schritt die Balancefaktoren für alle *relevanten* Knoten auf dem Pfad von  $x$  zur Wurzel an.

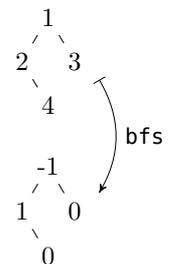


## Aufgabe 2 (AGS 3.2.49 c \*, AGS 3.2.33 c)

Gegeben ist die Typdefinition für Binärbäume:

```
typedef struct node *tree;
struct node { int key; tree left, right; };
```

- (a) Schreiben Sie eine C-Funktion `bfs`, welche für einen beliebigen Binärbaum  $t$  einen Binärbaum  $s$  zurückgibt, sodass  $s$  aus  $t$  hervorgeht, indem der Balancefaktor jedes Knotens dem Schlüsselwert dieses Knotens zugewiesen wird. Der Baum  $t$  soll dabei nicht verändert werden. Die Bäume rechts zeigen ein Beispiel des zurückzugebenden Baumes  $s$  (unten) für eine Eingabe  $t$  (oben).



- (b) Schreiben Sie eine Funktion `lRot`, die an der Wurzel eines übergebenen Baums eine einfache Linksrotation (im Sinne der AVL-Bäume) durchführt. Sie können davon ausgehen, dass der Wurzelknoten einen rechten Kindknoten hat.

### Aufgabe 3 (AGS 9.1.2 ★)

Gegeben sei der gerichtete Graph  $G = (V, E)$  mit der Knotenmenge  $V = \{1, \dots, 9\}$  und der Kantenmenge  $E = \{(1, 5), (2, 5), (3, 5), (4, 5), (5, 6), (6, 7), (6, 8), (6, 9)\}$ .

- Geben Sie eine graphische Darstellung von  $G$  an. Wie viele sequentielle Abarbeitungsreihenfolgen (topologische Sortierungen der Knoten) sind bei diesem Graphen möglich?
- Wie viele topologische Sortierungen bleiben übrig, wenn verboten wird, dass der Knoten 1 an erster Stelle steht?
- Verändern Sie in  $G$  die Kantenmenge  $E$  derart, dass der entstehende Graph  $G'$  genau die topologischen Sortierungen von  $G$  besitzt, die ausschließlich mit dem Knoten 1 beginnen.

### Zusatzaufgabe 1 (AGS 8.20 ★)

Fügen Sie in die folgenden AVL-Bäume den jeweils angegebenen Schlüssel ein. Stellen Sie nach jedem Einfügen die AVL-Eigenschaft her und dokumentieren Sie hierbei die ausgeführten Operationen wie üblich.

