

Informatik I

PD Dr. J.-G. Smaus
G. Röger, R. Mattmüller
Wintersemester 2010/2011

Universität Freiburg
Institut für Informatik

Übungsblatt Z

Abgabe: keine Abgabe (Zusatzaufgaben)

Aufgabe Z.1 (Suchbäume)

Eine Python-Implementierung von Suchbäumen und den wichtigsten Operationen darauf (Mitgliedschaftstest, Einfügen) finden Sie unter <http://www.informatik.uni-freiburg.de/~ki/teaching/ws1011/infoI/python/trees.py>.

- (a) Schreiben Sie eine Methode `inorder_traversal`, die eine Python-Liste mit den Elementen des Suchbaums zurückliefert, wobei immer gelten soll, dass zuerst die Elemente des linken Teilbaums (in `inorder_traversal`-Reihenfolge) kommen, dann das Datenelement am aktuellen Knoten, und schließlich die Elemente des rechten Teilbaums (in `inorder_traversal`-Reihenfolge).
- (b) Schreiben Sie eine Methode `preorder_traversal`, die eine Python-Liste mit den Elementen des Suchbaums zurückliefert, wobei immer zuerst das Datenelement am aktuellen Knoten kommt, dann die Elemente des linken Teilbaums (in `preorder_traversal`-Reihenfolge), und schließlich die Elemente des rechten Teilbaums (in `preorder_traversal`-Reihenfolge).
- (c) Schreiben Sie eine Methode `postorder_traversal`, die eine Python-Liste mit den Elementen des Suchbaums zurückliefert, wobei immer zuerst die Elemente des linken Teilbaums (in `postorder_traversal`-Reihenfolge) kommen, dann die Elemente des rechten Teilbaums (in `postorder_traversal`-Reihenfolge), und schließlich das Datenelement am aktuellen Knoten.
- (d) Schreiben Sie eine Methode `is_balanced`, die genau dann `True` zurückliefert, wenn der Baum im folgenden Sinne balanciert ist, und `False`, sonst: Ein binärer Baum ist genau dann balanciert, wenn für jeden Knoten p des Baumes gilt, dass sich die Höhe des linken Teilbaumes von p von der Höhe des rechten Teilbaumes von p höchstens um 1 unterscheidet.