Problem sheet 6

Course Fundamental Computer Science, Dr. Holger Kenn
e-mail: h.kenn@iu-bremen.de, tel.:+49 421 200 3112

1.) Rotations

We have seen that several balanced binary trees use rotations for the rebalance operations. Show that any arbitrary n-node binary tree can be transformed into any other arbitrary n-node binary tree using $O(n)$ rotations. (Hint: First show that at most $n-1$ right rotations suffice to transform the tree into a right-going chain.) (2P)

2.) Deletions

a) Is the operation of deletion ”commutative” in the sense that deleting x and then y from a binary search tree leaves the same tree as deleting y and then x? Argue why it is or give a counterexample. (2P)

b) When node z in TREEDELETE has two children, we could splice out its predecessor rather than its successor. Some have argued that a fair strategy, giving equal priority to presecessor and successor, yields better empirical performance. How might TREEDELETE be changed to implement such a fair strategy? (2P)

3.) Augmenting Datastructures

Can the black-heights of nodes in a red-black-tree be maintained as fields in the nodes of the tree without affecting the asymptotic performance of any of the red-black tree operations? Show how, or argue why not. Can the depth of nodes be maintained the same way? Show how, or argue why not. (4P)