



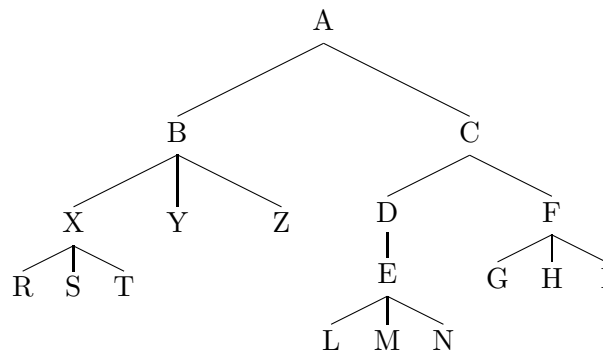
20 May 2008

Proseminar Algorithmen und Datenstrukturen

Exercise Sheet 8

Exercise 1 (Trees)

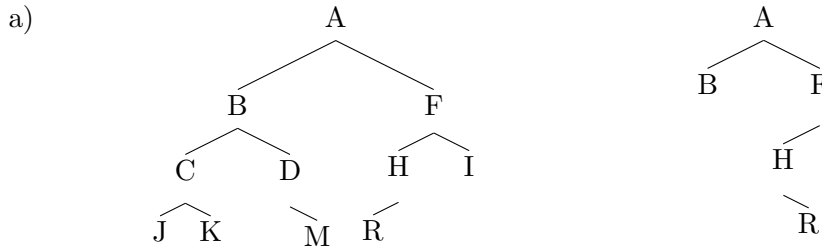
In the lecture the following terms are defined in the context of trees: node, edge, root, degree, level, height, ancestor, descendant, parent, child, path. Consider the tree given in the figure below and answer the following questions.



- Which node is the root node?
- Give the degree of the following Nodes: D, E, C. What is the degree of the tree itself?
- What is the level of the following nodes: A, B, Z, D, G, L? What is the height of the tree?
- Categorize the relation between the following pairs of nodes with, ancestor, descendant, parent, child: (Y,A),(Y,B)
- Give all child nodes, the parent node, all ancestors and all descendants of D.
- List all paths that contain Node D.

Exercise 2 (AVL-Trees)

For the following trees decide whether they are balanced or not and specify balance factors for all nodes. If a tree is not balanced identify all roots of smallest unbalanced subtrees and balance them using rotations.



- b) Insert the following nodes into an empty AVL tree. Show each step and what rotations are needed. Nodes to insert: 10, 40, 35, 25, 60, 30, 80, 50, 27, 28, 38
- c) Consider the record describing binary trees from the last exercise sheet. Modify the record so that it can be used to represent AVL trees.
- d) Assume that a new node was inserted into the AVL tree and that the balance factors have not yet been updated. Further assume that all keys are distinct. Use pseudo code to describe an algorithm that detects if the tree is in imbalance for the inserted node, and corrects the tree with rotations if necessary. Also update the balance factors.

The head of the algorithm should be as follows:

Listing 1 Rebalance

Input: T : pointer to the root of a tree which is an AVL-tree where a node with key key has just been inserted.

key : key of the inserted node.

Output: T : pointer to the restructured AVL tree.

Exercise 3 (Binary Search)

Extend the binary search example from the last exercise sheet such that it is able to work as a telephone book. Create a user interface that allows the user to insert, search and delete telephone numbers. As a simplification assume that each person is identified by the lastname. Also assume that telephone number and lastname are not longer than 50 characters.

Hint: For string comparison use the function `strcmp`. For string assignment use `strcpy`.