Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Dmytro Chibisov
Sandeep Sadanandan

Winter Semester 2007/08
Problem Sheet 8
December 17, 2007

Fundamental Algorithms

Deadline: December 19, 2007

ab/B - Trees

Problem 1

Explain *ab*-trees and B-trees. What are the differences?

Problem 2

For an *ab*-tree of height h (root node is at level zero) and n leaves, prove:

1. $2a^{h-1} \leq n \leq b^h$
2. $\lg_b(n) \leq h \leq \lg_a(\frac{n}{2}) + 1$

Problem 3

Starting from scratch, build a 2-3-tree with the following keys. [60, 76, 57, 48, 85, 55, 19, 56, 52]. Once the tree is created, delete the following elements from the tree. [48]

(Balanced) Binary (Search) Trees

Problem 4

Explain the three different traversals which could be done on a binary tree. Give the algorithem (pseudo) for all of them.

1. To print the elements of the Binary Search tree in a non-decreasing order, which traversal must be used?
2. In destroying a tree which traversal should be used? Why?

Problem 5

Show that if a node in a binary search tree has two children, then its successor has no left child and its predecessor has no right child. (Successor and Predecessor - both in-order)

Problem 6

Suppose n pair-wise unique keys are stored in a sorted array (ascending order). Let $Rang(k)$ be the index of key k in this sorted order. Consider the following implementation of `is_element(k)`:

$i = 0$;
while $2^i \leq n$ and $A[2^i] < k$ **do**
  $i++$;
end while
if $2^i \leq n$ **then**
  $binary\_search(A, 2^{i-1}, 2^i, k)$;
else
  $binary\_search(A, 2^{i-1}, n, k)$;
end if

Why does this approach work, and what is its complexity?

Problem 7

Suppose we implement an AVL tree with an additional pointer to the leftmost node (containing the smallest key). This allows us to execute a `find_min()` operation in constant time. Show how the implementations of the AVL tree operations can be modified to keep this pointer current, without increasing their asymptotic complexity by more than a constant factor.

Problem 8

Given a Binary Tree, how do you decide it's a binary search tree? Give explanation.

Problem 9

Describe an algorithm to select the k^{th} key in a binary search tree

Given a tree with n nodes, $k = 0$ selects the smallest key, $k = n - 1$ selects the largest key, and $k = \left\lfloor \frac{n}{2} \right\rfloor - 1$ selects the median key.