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

Winter Semester 2007/08
Problem Sheet 4
November 14, 2007

Fundamental Algorithms

Deadline: November 21, 2007

Problem 1 (15 Points)

The iteration operator "*" used in the $\lg^*$ function can be applied to monotonically increasing functions over the reals. For a function f satisfying $f(n) < n$, we define the function $f^{(i)}$ recursively for nonnegative integers i by

$$f^{(i)}(n) = \begin{cases} f(f^{(i-1)}(n)) & \text{if } i > 0, \\ n & \text{if } i = 0. \end{cases}$$

For a given constant $c > 0 \in \mathbf{R}$, we define the iterated function f_c^* by
$f_c^*(n) = \min\{i \geq 0 : f^{(i)}(n) \leq c\}$,
which need not be well-defined in all cases. In other words, the quantity $f_c^*(n)$ is the number of iterated applications of the function f required to reduce its argument down to c or less.

For each of the following functions $f(n)$ and constants $c > 0$, give as tight a bound as possible on $f_c^*(n)$.

	$f(n)$	c	$f_c^*(n)$
a.	$\lg n$	1	
b.	$n - 1$	1	
c.	$n/2$	1	
d.	$n/2$	2	
e.	$\sqrt{n}$	2	

Problem 2 (10 Points)

From the Mergesort algorithm explained in the class, Explain that the recurrence relation for the time cost of the algorithm is

$$T(n) = 2 \cdot T(\frac{n}{2}) + n$$

Prove that the cost is $O(n \lg n)$.

Instead of dividing the array into two parts, Would it be of any advantage if it is divided into three parts? What changes will have to be done in merging? Write down the recurrence relation.

Problem 3 (10 Points)

The Mergesort is an example for a paradigm called *Divide and Conquer*. The method is to divide the problem into smaller problems, solve them separately and then join the solutions.

Give another example for a *Divide and Conquer* algorithm.

Problem 4 (10 Points)

Using induction, develop an algorithm that finds the second largest number in a set of n (pairwise different) natural numbers.

Extra

This was already taken in the class. So this could be avoided. We will do this if time permits.

Problem 5 (10 Points)

Analyze the total number of operations (not only key-comparisons) executed when InsertionSort is applied to an input consisting of n keys. Write this resulting complexity in Landau notation and compare it to the number of key-comparisons analyzed in the lecture.