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

Winter Semester 2007/08
Problem Sheet 3
November 08, 2007

Fundamental Algorithms

Deadline: November 14, 2007

Problem 1 (5 Points)

Consider the definitions of o and ω given below.

$f(n) = o(g(n))$ iff $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$

$f(n) = \omega(g(n))$ iff $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$

From these definitions, derive the definitions of o and ω which were given in the class. (Just give an intuitive explanation)

Problem 2 (10 Points)

Let $f(n)$ and $g(n)$ be asymptoticallly positive functions. Prove or disprove the following.

1. $f(n) = O(g(n))$ implies $g(n) = O(f(n))$
2. $f(n) = O(g(n))$ implies $\lg f(n) = O(\lg g(n))$. Assume $\lg g(n) > 0$ and $f(n) \geq 1$ for all sufficiently large n.
3. $f(n) = O(g(n))$ implies $2^{f(n)} = O(2^{g(n)})$
4. $f(n) = O(g(n))$ imples $g(n) = \Omega(f(n))$

Problem 3 (10 Points)

Prove or disprove the following

1. $o(f(n)) \cap \omega(f(n)) = \phi$
2. $O(f(n)) - O(f(n)) = 0$

Problem 4 (15 Points)

Fill in the cells of the following table with "yes" or "no", depending on the relationships of functions $f(n)$ and $g(n)$. Other variables: $k \geq 1$, $\epsilon > 0$, $c > 1$ and $m > 1$ are constants.

$f(n)$	$g(n)$	O	o	Ω	ω	Θ
$\lg^k n$	n^ϵ					
n^k	c^n					
2^n	$2^{\frac{n}{2}}$					
$n^{\lg m}$	m^{lgn}					
$\lg(\lg^* n)$	$\lg^*(\lg n)$					

Note: $\lg^*$ is called the iterative logarithm function. It is defined as the number of successive applications of lg function on a given positive number n, until it reduces to 1.
Example: $\lg^* 16$.

$$\begin{aligned} \lg 16 &= 4 \\ \lg 4 &= 2 \\ \lg 2 &= 1 \end{aligned}$$

Here 3 calls of the function lg were possible before n reduced to 1. Hence, $\lg^* 16 = 3$.

Problem 5 (10 Points)

Write down the contents of the following arrays after every step of selection sort until they are completely sorted. Assume that the arrays given represent their initial arrangement of the numbers. Also compute the number of operations needed. (Comparison and Swapping are the operations)

1.

12	8	-2	23	5	0

2.

31	17	29	11	7	5	3