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

Winter Semester 2006/07
Problem Sheet 1
October 30, 2006

Fundamental Algorithms

Deadline: October 30, 2006

Problem 1 (10 Points)

Calculate the cost of calculating n^{th} Fibonacci number, using the recursive algorithm $F(n) = F(n-1) + F(n-2)$

Problem 2 (10 Points)

Show: $\left\lfloor 2^{\frac{n-1}{2}} \right\rfloor \leq F(n) \leq \left\lfloor 2^{\frac{n+1}{2}} \right\rfloor$

Problem 3 (10 Points)

Let SuperComputer be a very fast computer which can perform 10^9 operations per second, For some problems of size n the table below lists the number of operations necessary. More specifically, the i^{th} algorithm needs $t_i(n)$ operations.

$$\begin{aligned}
t_1(n) &= 2 \cdot n \\
t_2(n) &= n \lg(n) \\
t_3(n) &= 2.5n^2 \\
t_4(n) &= \frac{1}{1000} \cdot n^3 \\
t_5(n) &= 3^n
\end{aligned}$$

Determine, for which maximal input sizes each algorithm needs at most 1 second, 1 minute, 1 hour.
How do these values change, if the computer is upgraded to be 10 times faster (i.e., can do 10^{10} operations)?

Problem 4 (20 Points)

Design iterative and recursive algorithms to compute 2^n. Show that there exists a recursive algorithm which performs better than the iterative naive algorithm.