Technische Universität München
Institut für Informatik
Lehrstuhl für Effiziente Algorithmen
Lehrstuhl für numerische Programmierung

WS 2002/2003
Worksheet 1
17.10.2002

Fundamental Algorithms

Exercise 1

Consider two different algorithms, $\mathcal{A}$ and $\mathcal{B}$, that solve the same problem. $\mathcal{A}$ shall require $500n^2 - 16n$ operations for a problem of size n; $\mathcal{A}$ shall require $\frac{1}{3}n^3 + \frac{11}{2}n + 7$ operations for the same problem.

(a) If you had a problem of size $n = 256$, which algorithm would you choose?

(b) Which algorithm would you choose, if you had a problem of size $n \geq 2000$?

Exercise 2

HOMECOMPUTER shall be a machine that can perform 10^9 operations per second. Consider that we have five different algorithms for a specific problem. For each algorithm i, we know the number of operations $T_i(n)$ it will perform on a problem of size n:

$$\begin{aligned} T_1(n) &= 6\,000\,000 \cdot n \\ T_2(n) &= 60\,000 \cdot n \log n \\ T_3(n) &= 0.003 \cdot n^2 \\ T_4(n) &= 10^{-6} \cdot n^3 \\ T_5(n) &= 10^{-18} \cdot 2^n \end{aligned}$$

For each algorithm compute the size $n_{\max}$ of the largest problem the respective algorithm can solve within 1 second (1 minute, 1 hour, ...). Enter the maximal problem sizes into the following table:

	1s	1m (60s)	1h (3600s)	1d (86 400s)	30d ($\approx 2.6 \cdot 10^6$s)	1a ($\approx 3.2 \cdot 10^7$s)
T_1						
T_2						
T_3						
T_4						
T_5						

Exercise 3

Write a program for a RAM that will compute the n-th Fibonacci number F_n. Specify the starting configuration (esp. in which register the input n should be placed), and determine the number of work cycles the RAM will perform on the input n (i.e. the uniform time complexity of the RAM).

It is recommended to use the **iterative** algorithm for the Fibonacci numbers as a basis for the RAM program.