Technische Universität München
Institut für Informatik
Theoretische Informatik

WS 2015/16
Worksheet 2
26.10.2015

Fundamental Algorithms

Exercise 1

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						

K-Exercise 2 (MergeSort)

Compute the number of comparisons that will be performed by MergeSort in the best case (i.e., compute this number exactly, including the constants for $\Theta(n \log n)$.

Note:

A K-Exercise*, in this course, will mark an exercise that might (in a similar form) well occur in the exam.*

Exercise 3 (Sorting)

Prove or disprove the following statement – or try at least to figure out whether it holds or not:
If we sort each row of a matrix, and, after that, sort each column of the matrix, the rows of the matrix will still be sorted afterwards.