Technische Universität München
Institut für Informatik
Theoretical Computer Science

WS 2017/18

Fundamental Algorithms 2

Exercise 1 (Real Complexity)

Suppose HomeComputer is 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:

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

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:

	1 second	1 minute	1 hour	1 day	1 month (30 d)	1 year (365 d)
T_1						
T_2						
T_3						
T_4						
T_5						

Exercise 2 (MergeSort)

Compute the number of comparisons between array elements that will be performed by MergeSort on an array of size $n = 2^k$, $k \in \mathbb{N}$ in the best case, i.e., compute this number exactly.

Exercise 3 (Sorting)

Prove or disprove the following statement: 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.