Technische Universität München
Department of Computer Science
Chair for Efficient Algorithms
Prof. Dr. Sami Khuri

WS 1999/2000
Problem Set 3
November 16, 1999

Efficient Algorithms and Data Structures

Problem 1 (9 credits)

Solve the recurrence:

$$\begin{aligned} T(1) &= 2 \\ T(n) &= n + 1 + \sum_{j=1}^{n-1} \frac{(2j+1)T(j)}{n^2}, \ n \geq 2 \end{aligned}$$

What is the growth rate of $T(n)$? (In other words, find the simplest function $f(n)$ such that $T(n)$ is $\Theta(f(n))$.)

Problem 2 (9 credits)

Solve the recurrence:

$$\begin{aligned} A(1) &= 1 \\ A(n) &= \frac{1}{n}\left(\sum_{i=1}^{n-1} A(i) + \frac{n^2+3n-2}{2}\right), \ n \geq 2 \end{aligned}$$

Problem 3 (8+8 credits)

Suppose we design a divide-and-conquer algorithm for matrix multiplication based on partitioning an $n \times n$ matrix into 9 submatrices of size $\frac{n}{3} \times \frac{n}{3}$. This gives rise to a recurrence:

$$T(n) = a \cdot T(n/3) + c \cdot n^2 \tag{1}$$

a) What is the value of a, if "ordinary" matrix multiplication is performed? Solve the recurrence exactly for n a power of 3, with starting value $T(1) = 1$. Prove your result by induction. What is the growth rate of $T(n)$?

b) What is the largest (integer) value of a in the original recurrence (1) that would lead to a growth rate less than Strassen's algorithm? (Note: Strassen's algorithm has growth rate $\Theta(n^{\log_2 7})$.)

Problem 4 (6 credits)

How would you modify Strassen's algorithm to multiply $n \times n$ matrices in which n is not an exact power of 2? What is the growth rate of the resulting algorithm?