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

WS 1999/2000
Problem Set 5
November 30, 1999

Efficient Algorithms and Data Structures

Problem 1 (8 credits)

Use generating functions to solve:

a) $a_n = 5a_{n-1} - 6a_{n-2}, \qquad a_0 = 1,\ a_1 = -2$

b) $a_n = 2a_{n-1} + a_{n-2} - 2a_{n-3}, \qquad a_0 = 1,\ a_1 = 2,\ a_2 = 0$

Problem 2 (6 credits)

It is said that the origin of the Towers of Hanoi problem is as follows. After creating the world, God set on Earth three rods made of diamond and 64 rings of gold. These rings are all different in size. At the creation they were threaded on the first rod in order of size, the largest at the bottom and the smallest at the top. God also created a monastery close by the rods. The monks' task in life is to transfer all the rings onto the second rod. The only operation permitted is to move a single ring from one rod onto another in such a way that no ring is ever placed on top of another smaller one. When the monks have finished their task, according to the legend, the world will come to an end. (This is probably the most reassuring prophecy ever made concerning the end of the world, for if the monks manage to move one ring per second, working night and day without ever resting nor ever making a mistake, their work will still not be finished 500000 million years after they began. This is more than 25 times the estimated age of the Universe!)

Let $T(n)$ represent the number of moves required to solve the Tower of Hanoi problem with n rings, $n > 0$.

a) Find a recurrence relation for $T(n)$.

b) Find the closed form of $T(n)$ by two different methods.

Problem 3 (6 credits)

Consider a 1 by n chessboard. Suppose that we color each square of the chessboard in either blue or red. Let $T(n)$ represent the number of possible colored chessboards in which no two squares that are colored red are adjacent.

a) Find a recurrence relation for $T(n)$.

b) Find the closed form of $T(n)$ by two different methods.

Problem 4 (6 credits)

Let $T(h, k)$ be the maximum number of leaves of a tree of height h, where each node has outdegree (number of children) k or less.

a) Find a recurrence relation for $T(h, k)$.

b) Solve the recurrence relation by two different methods.

Problem 5 (6 credits)

Use generating functions to show that the solution of the recurrence relation

$$a_n = \left(\frac{2}{n}\sum_{k=0}^{n-1} a_k\right) + cn + d, \qquad n \geq 1$$

is

$$a_n = 2c(n+1)H_n + (a_0 - 3c + d)n + a_0,$$

where c and d are constants.

Note that the above recurrence relation is often encountered when analyzing the amount of work involved in an algorithm of the "decomposition type." See Section 1.4 and Section 3.2.2 (pages 83–85) in "Combinatorial Algorithms: Theory and Practice," by E. Reingold, J. Nievergelt and N. Deo, for more information

Hint: This problem is quite challenging. At some point, you might find the following substitution quite useful:

$$G(x) = \frac{F(x)}{(1-x)^2}$$

Problem 6 (8 credits)

Consider the problem of making change for n cents using the least number of coins.

a) Describe a greedy algorithm to make change consisting of quarters (25 cents), dimes (10 cents), nickels (5 cents), and pennies (1 cent). Prove that your algorithm yields an optimal solution.

b) Suppose that the available coins are in the denominations c^0, c^1,...,c^k for some integers $c > 1$ and $k \geq 1$. Show that the greedy algorithm always yields an optimal solution.

c) Give a set of coin denominations for which the greedy algorithm does not always yield an optimal solution.