Advanced Algorithms, Fall 2011

Prof. Bernard Moret

Homework Set #1

Your solutions (PDF files) are due (email to `lcbb.epfl@gmail.com`) on Monday, Oct. 3, by midnight.

Question 1. Using any of your favorite solution methods, solve the following recurrences in Θ terms.

1. $f(n) = f(n/2) + 2f(n/4) + \Theta(n)$
2. $f(n) = f(n-1) \cdot f(n-2)$, with $f(1) = 1$ and $f(2) = 2$
3. $f(n) = \sum_{i=1}^{n-1} f(i)$ with $f(1) = 1$
4. a bit harder: let $f(n) = f(n-1)/2 + 2/f(n-1)$, with $f(0) = 3$, and define $g(n) = \Pi_{i=1}^{n} f(i)$ (hint: guess what the answer may be and verify it using limits of ratios)

Question 2. In mathematics, one can write all sorts of complicated things with very simple notation; occasionally, some of these complicated things may even be expressible in a simpler way. So solve the following very complicated recurrence, which has a *very* simple solution:

$$f(x^2 + 4f(x) - 1) = f^2(x+1)$$

Question 3. You are given an $N \times N$ matrix of distinct integers, sorted in increasing order along every row and along every column. You are also given a particular integer x that is present in the matrix. Devise and analyze an algorithm that, for any such matrix and any such integer, locates the row and column where this element is to be found. You should base your algorithm on a binary search idea: get the "middle" element (initially, the element at row and column $N/2$), compare it with x, then search recursively in the appropriate three of the four $N/2 \times N/2$ submatrices around that midpoint.

Question 4. Define the following approach for the multiplication for length-n binary vectors.

If we have $A = A[1 \ldots n] = A_L A_R$ and $A_L = A[1 \ldots n/2]$, $A_R = A[n/2+1 \ldots n]$, then we can write $XY = 2^n X_L Y_L + 2^{n/2}(X_L Y_R + X_R Y_L) + X_R Y_R$.

Now use this divide-and-conquer approach recursively until each vector has one element. Analyze the number of multiplications in this approach. Can you improve this approach? (Hint: $ab + cd = (a+c)(b+d) - ac - bd$)

Question 5. Matrix multiplication is easy to break into subproblems. For example, if we are given $A = \begin{vmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{vmatrix}$ and $B = \begin{vmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{vmatrix}$
then we can compute the product AB as follows:
$AB = \begin{vmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{vmatrix}$
(A and B are $(n \times n)$ matrices, and A_{ij} and $B_{i,j}$ are $(n/2 \times n/2)$ matrices.) Now use this divide-and-conquer approach recursively until each block has one element. Analyze the resulting number of multiplications.