Algorithms – CMSC-27200/37000 Homework – January 9, 2003

Instructor: László Babai Ry-164 e-mail: laci@cs.uchicago.edu

Convention. In all assignements, the term "grad students" will refer to those seeking 37000 credit, regardless of actual graduate or undergraduate status, similarly, "undergraduates" refers to those seeking CS-27200 credit. **G** and **U**, respectively, will abbreviate these terms.

READING (due next class) Handouts: 1. Divide and Conquer: The Karatsuba–Ofman algorithm for multiplication of large integers. 2. Asymptotic notation. (Undergraduates may ignore the "partition function" $p(n)$ discussed in that handout.)

Review mathematical induction, quantified formulas, asymptotic notation from Discrete Math. Find and study examples of PSEUDOCODES in the text.

Grad students: find in the text and learn Strassen's algorithm for matrix multiplication.

HOMEWORK. Please **print your name and U/G status on each sheet.** Please try to make your solutions readable. Unless expressly stated otherwise, all solutions are due at the **beginning of the next class.**

2.1 (U, G) (8 points) *(The complexity of matrix multiplication.)* An $n \times n$ matrix is an $n \times n$ array of numbers. Let A and B be two $n \times n$ matrices. The product of A and B is defined as the $n \times n$ matrix $C = AB$ where

$$C[i,j] := \sum_{k=1}^{n} A[i,k]B[k,j].$$

Let us consider the **model** where multiplication of numbers costs one unit, addition, subtraction, and bookkeeping are free. In this model, the cost of multiplying two $n \times n$ matrices is $O(n^3)$.

In 1969, *V. Strassen* found a surprising way to reduce the multiplication of two $n \times n$ matrices to multiplying seven pairs of $n/2 \times n/2$ matrices, and doing some additions and subtractions (which are free in our model). Using this, a less expensive recursive algorithm for matrix multiplication follows.

Analyse Strassen's algorithm in the spirit of the analysis of the Karatsuba–Ofman algorithm (handout).

Let $S(n)$ denote the cost of multiplying two $n \times n$ matrices via Strassen's algorithm. Assume $n = 2^k$. **State a recurrence** for $S(n)$. Solve the recurrence. Your answer should be of the form $S(n) = n^\beta$. Determine the exact value of the constant β (by a formula) and calculate β to 4 significant digits of accuracy. (OVER)

2.2 (U,G) (4 points) Prove: if p is a prime number and $p > 3$ then $p \equiv \pm 1 \pmod 6$. (Your proof should be very short. Clarity counts.)

A REQUEST: Please write your solutions to "G only" problems on a separate sheet and put them on a separate pile when you hand them in.

Undergraduates: do READ the "grad only" problems; receive bonus points for solving them.

2.3 (G only) **Undergraduates: study, understand, and remember the important theorem stated in this problem.** (6 points) Let a_n, b_n be sequences of positive real numbers. Prove: if $\lim_{n\to\infty} a_n = \infty$ and a_n is $\Theta(b_n)$ then $\ln a_n \sim \ln b_n$.

2.4 (G only) (6 points) **(Generating random prime numbers)** Suppose we can test primality of $\le n$-bit integers (i. e., decide whether or not a given integer with $\le n$ binary digits is prime) in $f(n)$ computational steps in a reasonable model of computation. Prove that a random $\le n$-bit prime number can be selected in expected $O(n^2 f(n))$ steps, where the steps permitted include flipping a fair coin. (The prime number selected must be uniformly distributed over all prime numbers with $\le n$ bits.) *Hint.* Use the Prime Number Theorem (see "Asymptotic Notation" handout).