Algorithms – CS-27200/37000   Homework – January 12, 2004
Instructor: László Babai   Ry-164   e-mail: laci@cs.uchicago.edu

ADVICE. Take advantage of the TA's office hours.

READING (due next class)

Evaluation of recurrent inequalities; Asymptotic notation; Divide and Conquer: the Karatsuba–Ofman algorithm;

Review mathematical induction, quantified formulas, asymptotic notation from Discrete Math.

HOMEWORK. Please **print your name on each sheet.** Print "U" next to your name if you seek 27200 credit and "G" if you seek 37000 credit, regardless of your grad/undergrad status. Please try to make your solutions readable. Unless expressly stated otherwise, all solutions are due at the **beginning of the next class.**

3.1 (U,G) A "problem instance" of size n means an input of size n for the given computational task. A *recursive algorithm* reduces a problem instance to one or more smaller instances of the same problem. As we have seen in class, the analysis of such an algorithm leads to a recurrence.

(a) (8 points) Suppose a recursive algorithm reduces an instance of size n to four instances of size $n/3$ each. Let $R(n)$ be the number of operations required for inputs of size n. Describe the recurrence for $R(n)$ and evaluate it. Ignore the cost of the reduction. Assume that $n = 3^k$ and $R(1) = 5$.

Your answer should express $R(n)$ in the form $R(n) = a \cdot n^b$ where a and b are positive constants. You need to find the values of a and b. Describe the exact values of a, b and calculate them to 3 significant digits of accuracy.

(b) (G only; 8 points) We modify part (a): we no longer ignore the cost of the reduction but assume instead that the cost of the reduction is $O(n)$. Prove that $R(n) = O(n^c)$ for some constant c. Determine the smallest possible value of c exactly (with a formula) and to 4 significant digits of accuracy. Use the method of "reverse inequalities" from the handout.

3.2 (G only) (8 points; please write this solution on a separate sheet and put it on a separate pile when you hand it in.) A recursive algorithm reduces an instance of size n to n instances of size $n/2$. Let $S(n)$ be the number of operations required for inputs of size n. Describe the recurrence for $S(n)$ and evaluate it. Ignore the cost of the reduction. Assume that $n = 2^k$ and $S(1) = 1$.

Your answer should express $S(n)$ in the form $S(n) = a \cdot n^{b+c \log n}$ where a, b, c are positive constants. You need to find the values of a, b, c.

3.3 (U,G) (4 points) In what follows, try to get *reasonable* estimates for "problem size" of the instances below. Give **short** justification for the assumptions you make, and indicate what sources you used for your estimates. Input size is measured in number of bytes.

1. Student records of current UofC students.
2. Census data for the US census.
3. Bibliographic records of all holdings at Regenstein Library.
4. DVD of the Twin Towers part of the Lord of the Rings movie.
5. Data captured over a 1-week period by a typical EOS satelite.