Algorithms – CMSC-37000

Instructor: László Babai Ryerson 164 e-mail: laci@cs.etc

Homework set #1 due January 14, 2014

Please print your name, major/field, year, Undergrad/Graduate status on your homework. The problems marked "HW" are homework problems due Tuesday, January 14, **before class.** (Note: there are two of these; only one of them was assigned in class.) "Challenge problems" are optional, do not count toward the grade, and have no deadline except they expire when discussed in class or in the tutorial. If you hand in a solution to a Challeng problem, please do so on a *separate sheet* – these problems are graded separately. The "DO" exercises are meant to be solved for your own benefit by the same deadline as the HW; do not hand in the solution. Solving the "DO" exercises is important to help you internalize the concepts.

1.1 ("DO") Review **asymptotic notation** from Discrete Math, Fall 2014. Use the instructor's online lecture notes as well as the problem sets posted.

1.2 (**HW**, due Jan 14) Let $b(n)$ denote the number of binary digits of the positive integer n and let $d(n)$ denote the number of its decimal digits. (To make this definition unique, no initial zeros are allowed in this problem.) (a) Give a simple exact formula for $b(n)$ using the logarithm function and rounding. (b) Prove the asymptotic relation $b(n) = \Theta(d(n))$. **(3+3 points)**

1.3 ("DO" exercise, do not hand in.) Recall the communication complexity problem discussed in class. Use variables where we used specific numbers. When talking about k-bit itegers, we permit initial zeros, so strictly speaking, we are talking about integers with $\leq k$ bits, i. e., integers between 0 and $2^k - 1$.

Here is the setup.

Two processors, Alice and Bob, possess a string of n bits each; Alice's string is X, Bob's string is Y. The problem is to determine whether or not $X = Y$ with minimum number of bits communicated between Alice and Bob. The randomized protocol discussed in class to solve this problem efficiently depends on the choice of a parameter k and proceeds in the following steps:

1. Alice generates a k-bit prime, chosen uniformly at random from among all k-bit prime numbers. (Each k-bit prime has the same probability to be selected.)

2. Alice calculates the quantity $(X \bmod p)$, the remainder of the division of X (an n-bit integer) by p. Note that this remainder has at most k bits.
3. Alice sends p and $(X \bmod p)$ to Bob.
4. Bob calculates $(Y \bmod p)$.
5. If $(X \bmod p) \neq (Y \bmod p)$, i. e., if $X \not\equiv Y \pmod p$, then Bob says "NOT EQUAL." Else, Bob says "YES, EQUAL."

The cost of this protocol is at most $2k$ bits of communication (step 3); the cost of local computation by either Alice or Bob is ignored in the "communication complexity" model. We need to show that even for quite small values of k, Bob is not likely to make an error.

(a) Let Z be an n-bit non-zero integer. Let $p(Z)$ denote the number of distinct primes dividing Z. Prove: $p(Z) < n$.

(b) (CHALLENGE) Prove: $p(Z) \lesssim n/\log_2 n$. Here $a_n \lesssim b_n$ means $a_n \sim \min\{a_n, b_n\}$.

(c) Let R denote the probability that Bob's conclusion is wrong. Use (a) to prove that $R = O(kn/2^k)$.

(d) Use (b) to prove that $R = O(n/2^k)$.

(e) Given an error-tolerance parameter $\epsilon > 0$, use (d) to recommend a value of k as a function of n such that $R \leq \epsilon$. Make k as small as you can.

1.4 (**HW**, due Tuesday, Jan 14) Consider the situation that Bob declared "NOT EQUAL." At this point Alice possesses the n-bit integer X, Bob the n-bit integer Y, both of them have the k-bit prime p and the k-bit integer $(X \bmod p)$ and they know that $X \not\equiv Y \pmod p$.

Devise a *deterministic* protocol which uses $O(k \log n)$ bits of communication and finds a position i such that $X[i] \neq Y[i]$. ($X[i]$ denotes the i-th bit of X.) Describe your protocol in pseudocode. Prove that it works within the stated amount of communication. Do not make assumptions on the value of k relative to n. **(8 points)**