KTH Computer Science and Communication

Homework I, Complexity Theory Fall 2011

Due on September 8 at 15.15, i.e. at the beginning of the lecture. The general rules on homework solutions available at the course home-page apply. In particular, discussions of ideas in groups of up to at most two people are allowed but solutions should be written down individually.

Some of the problems below are "classical" and hence their solutions is probably posted on the Internet. It is not allowed to use such solutions in any way. The order of the problems is "random" and hence do not expect that the lowest numbered problems are the easiest.

Any corrections or clarifications on this problem set will be posted under "homework" on the course home-page http://www.csc.kth.se/utbildning/kth/kurser/DD2446/kplx11/uppgifter.

1 (10p) A Mersenne prime is a number on the form $2^n - 1$ that is prime. Today 47 such primes are known, the largest being $2^{43112609} - 1$ and it is believed, but unknown, whether there are infinitely many such primes. Consider the function $f(s)$ defined to be one if there is an $n > s$ such that $2^n - 1$ is a Mersenne prime and which is 0 otherwise. Determine whether f is a mechanically computable function.

2 Let M_x be the x'th "machine" of your favorite formal model (e.g. Turing machine or idealized C-program) and let $f(x)$ be the maximal number of computational steps taken by M_x on any input smaller than x *for which it halts* (and to be 0 if M_x is not a legal machine).

Let $g(y) = \max_{x \leq y} f(x)$. Your task is to try to make a good estimate of quickly $g(y)$ increases.

2a (10p) Can you make a lower bound by giving an explicit small program that runs for a long time on a small input?

2b (10p) Can you prove any upper bound on g? Be warned that this is a slightly open ended problem.

3 (10p) Make a (rough) sketch of how to simulate a two-tape Turing machine (find a definition of this machine on the web if you do not have the book) in your favorite programming language (do not write code, just pseudo-code). In particular address the question if and in such a case how you need to extend your programming language compared to its ordinary implementation on an ordinary 64-bit ordinary computer to make the simulation possible.