Prof. Friedrich Eisenbrand
Yuri Faenza

Discussions from: March 19, 2013

Computer Algebra

Spring 2013

Assignment Sheet 3

Exercises marked with a $\star$ can be handed in for bonus points. Due date is April 09.

Exercise 1

Show that the following alternative algorithm for computing the gcd of $a, b \in \mathbb{N}$ is correct and give an upper bound on its running time.

INPUT: $a, b \in \mathbb{N}$, OUTPUT: gcd(a,b).
SET $r = a$, $r' = b$, $e = 0$.
WHILE $2|r$ and $2|r'$

  SET $r = r/2$, $r' = r'/2$, $e = e + 1$.

WHILE $r' \neq 0$

  WHILE $2|r$ SET $r = r/2$.

  WHILE $2|r'$ SET $r' = r'/2$.

  IF $r' < r$ SET $(r, r') = (r', r)$.

  SET $r' = r' - r$.

RETURN $r \cdot 2^e$.

Exercise 2

Show that, if $a, p \in \mathbb{N}$ such that $a^p - 1$ is prime, then $a = 2$ or $p = 1$.

Exercise 3

Recall that an algorithm is said *polynomial time* if its running time is polynomial in the length of the input. Show that a polynomial time algorithm for the following problem (P):

INPUT: a pair of positive integers $\ell \leq N$.
OUTPUT: 'YES' if N has a prime factor greater than ℓ ; 'NO' otherwise.

implies the existence of a polynomial time algorithm for the following problem (F).

INPUT: a number N, OUTPUT: a factorization of N in prime numbers.

Exercise 4

A *proper factor* of an integer N is a number distinct from N that divides N. Show that a polynomial time algorithm for the following problem (P'):

```
INPUT: a pair of positive integers ℓ ≤ N.
OUTPUT: 'YES' if N has a proper factor greater than ℓ ; 'NO' otherwise.
```

implies the existence of a polynomial time algorithm for (F) (see Exercise 3).

Exercise 5
Suppose you *know* the answer to a problem for a given input. How do you convince someone that you are right? (In general, this task can be much easier than *computing* the correct answer). This idea is captured by the following definitions. A *YES-instance* (resp. *NO-instance*) for (P) is a pair ℓ, N such that the answer to problem (P) with input ℓ, N is YES (resp. NO). We say that $S : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ is a *YES-certificate* (resp. *NO-certificate*) for (P) if:

- $S(\ell, N)$ is of size polynomial in $\log(N)$;
- there exists a polynomial time algorithm that, given N, ℓ and $S(\ell, N)$ as an input, answers YES (resp. NO) if and only if ℓ, N is a YES- (resp. NO-) instance.

Give YES- and NO-certificate (and the corresponding algorithms) for the problem (P). You can assume that checking whether a number is prime can be performed in polynomial time.

Exercise 6
Give YES- and NO-certificate (and the corresponding algorithms) for the problem (P'). You can assume that checking whether a number is prime can be performed in polynomial time.

Exercise 7 ($\star$)
Prove that N is Carmichael if and only if N is composite, odd, each of its prime factors have multiplicity exactly one in its factorization and $p - 1 | N - 1$ for all primes $p | N$.

Exercise 8
Let $N = \prod_{i=1}^{k} p_i$ be a Carmichael number, with $p_1, \ldots, p_k$ primes. What is the probability that the Fermat Test will answer "composite" when N is given as an input? Assume that the Fermat test picks a random number between 1 and $N - 1$.

Exercise 9 ($\star$)

(a) Implement the Fermat Test and the Miller-Rabin test.

(b) Implement an algorithm that receives as input two prime numbers p and q and computes public and private keys as in the RSA cryptosystem.

(b) Implement an algorithm that receives as input the public and private keys of the RSA algorithm and an encrypted message and outputs the original message.

(c) Test the previous algorithm on the following inputs: public key: (43,9379); private key: (2563,9379); encrypted message: 2982. What is the original message?