Bordeaux 1 University
Mathematics

MHT 933 – Master 2
Year 2009-2010

Final Exam. 2009 December 14th, 14h-18h

Handwritten lecture notes are allowed as well as the course typescript. You may compose in either English or French.

Exercise 1 – [Proth's Theorem]
Let $n > 1$ be an odd integer. Then n can be written $n = s \cdot 2^r + 1$ with $s > 0$ odd, and $r > 0$. Proth's original theorem (1878) is the following one.
Theorem. *If $s < 2^r$ and if there exists an $a \in \mathbb{Z}$ such that*

$$a^{(n-1)/2} \equiv -1 \bmod n,$$

then n is prime.

We are first going to prove a stronger version of this result, replacing the condition $s < 2^r$ by $s < 2^{r+1} + 3$.

1) Suppose that $s < 2^{r+1} + 3$ and that such an a exists. Let p a prime divisor of n. By considering the order of a modulo p, show that $p \equiv 1 \bmod 2^r$.

2) Show that if n is composite, this implies that we have $s \geq 2^{r+1} + 3$ and conclude.

3) Now admit that n is as above and that we know an a such that $(\frac{a}{n}) = -1$ (Jacobi symbol). Give a deterministic and very simple algorithm which allows to know whether n is prime or composite.

4) What is the word complexity of this algorithm?

Exercise 2 – [Multipoint evaluation]
Let R a commutative ring and $m_0, \ldots, m_{n-1}$ in $R[X]$, non-constant, where $n = 2^k$. For $0 \leq i \leq k$, and $0 \leq j < 2^{k-i}$, define

$$M_{i,j} = \prod_{0 \leq l < 2^i} m_{j2^i + l}.$$

1) In the special case $n = 8$, write down a natural tree whose vertices at level i are labelled by the $M_{i,j}$, $j = 0, \ldots, 2^{3-i} - 1$.

2) Compute all $M_{i,j}$ in $\widetilde{O}(\sum_{i<n} \deg m_i)$ basic operations in R (recall that for $A, B \in R[X]$ we can compute AB in $\widetilde{O}(\deg A + \deg B)$ operations in R).

3) When all m_i have degree 1, compare with the naive algorithm which would only compute $M_{k,0}$ with successive multiplications by a factor of degree 1.

4) Let $T \in R[X]$ of degree $< n = 2^k$ and $u_0, \ldots, u_{n-1} \in R$. Let $m_i = X - u_i$ and assume that all $M_{i,j}$ are precomputed. Show that the following algorithm compute $T(u_0), \ldots, T(u_{n-1})$ in $\widetilde{O}(n)$ operations in R.

Algorithm 1. Multipoint evaluation

If $n = 1$ return T.
Let $r_0 \leftarrow T \operatorname{rem} M_{k-1,0}$. Compute recursively $r_0(u_0), \ldots, r_0(u_{n/2-1})$.
Let $r_1 \leftarrow T \operatorname{rem} M_{k-1,1}$. Compute recursively $r_1(u_{n/2}), \ldots, r_1(u_{n-1})$.
Return the concatenation of the outputs.

5) Show that a polynomial of arbitrary degree $< n$ can be evaluated at n points in $\widetilde{O}(n)$ operations in R. Compare with successive applications of Horner's scheme. Compare with the FFT algorithm.

Exercise 3 –[Pollard's and Strassen's method]

We shall study here an algorithm which, thanks to multipoint evaluation (exercice 2) factors an integer N which is neither a prime nor a perfect power in $\widetilde{O}(N^{1/4})$ word operations.

Let $N > 1$ be a composite integer which is not a perfect power and denote respectively by $S_1(N)$ and $S_2(N)$ the largest prime factor of N and the second largest prime factor of N. We have

$$S_2(N) < S_1(N) \quad \text{and} \quad S_2(N) < N^{1/2}.$$

We denote by $a \longmapsto \overline{a}$ the reduction of integers modulo N. The Pollard's and Strassen's factoring algorithm is the following one.

Algorithm 2. Pollard and Strassen

Require: $N \geq 6$ neither a prime nor a perfect power and $b \in \mathbb{N}$.
Ensure: The smallest prime factor of N if it is less than b, or otherwise `failure`.

$c \leftarrow \lceil b^{1/2} \rceil$ and compute the coefficients of $f(X) = \prod_{1 \leq j \leq c}(X + \overline{j}) \in (\mathbb{Z}/N\mathbb{Z})[X]$ thanks to the previous exercise.
Use the fast multipoint evaluation algorithm to compute $g_i \in \{0, \ldots, N-1\}$ such that $g_i \bmod N = f(\overline{ic})$ for $0 \leq i < c$.
if $\gcd(g_i, N) = 1$ for $0 \leq i < c$ **then**
    Return `failure`
else
    $k \leftarrow \min\{0 \leq i < c;\ \gcd(g_i, N) > 1\}$
    Return $\min\{kc + 1 \leq d \leq kc + c;\ d \mid N\}$.

1) Prove the correctness of the algorithm.

2) Prove that the algorithm works in $O(M(b^{1/2})M(\log N)(\log b + \log\log N))$ word operations where M is the multiplication time. Recall that a gcd computation of integers of length less than n can be done in $O(M(n)\log n)$ word operations and that a division with remainder of such integers can be done in $O(M(n))$ word operations.

3) Running the algorithm for $b = 2^i$ and $i = 1, 2, \ldots$, show that we can completely factor N in $\widetilde{O}(N^{1/4})$ word operations.

Exercise 4 – [Square roots in $\mathbb{F}_p$ and Cornacchia's algorithm]

Let $p = 2^e q + 1$ be an odd prime (where $e \geq 1$ and q is odd), and let $a \in \mathbb{F}_p^*$ a quadratic residue modulo p. We want to solve $x^2 \equiv a \mod p$.

1) Show that if $p \equiv 3 \bmod 4$, $x = a^{(p+1)/4} \bmod p$ is a solution. Prove also that if $p \equiv 5 \bmod 8$, either $x = a^{(p+3)/8} \bmod p$ or $x = 2a \cdot (4a)^{(p-5)/8} \bmod p$ is a solution.

Unfortunately, when $p \equiv 1 \bmod 8$ the problem is harder. Tonelli's and Shanks' algorithm solves it in all cases.

Algorithm 3. Tonelli and Shanks

1: Find an u which is not a quadratic residue modulo p (pick uniformly at random elements in $\{1, \ldots, p-1\}$ until we are satisfied). Then put $z \leftarrow u^q \mod p$.
2: Initialization : $k \leftarrow e$, $x \leftarrow a^{(q+1)/2} \mod p$, $b \leftarrow a^q \mod p$.
3: Determine the smallest m such that $b^{2^m} \equiv 1 \mod p$.
4: Put $t \leftarrow z^{2^{k-m-1}}$, $z \leftarrow t^2$, $b \leftarrow bz$ and $x \leftarrow xt$, the four affectations being done modulo p.
5: **if** $b = 1$ **then**
6:     Return x
7: **else**
8:     Put $k \leftarrow m$ and go back to 3.

2) What is the probability to be successless at step 1 after k successive trials?

3) Show that at each of the following steps we have $ab \equiv x^2 \mod p$ and that, if the algorithm terminates, we have a suitable x.

4) Show that the algorithm terminates, using at most e loops (have a look at the orders of b and z modulo p).

5) Show that the number of modular multiplications done after step 1 is in $O(\log q + e^2)$.

Let now p be a prime number and d an integer such that $0 < d < p$. We are searching for integers x and y such that

$$x^2 + dy^2 = p,$$

if they exist.

6) Show that, if the equation has solutions, then $-d$ is a quadratic residue modulo p.

Cornacchia's algorithm consists in determining an x_0 such that $0 < x_0 < p$ and $x_0^2 \equiv -d \mod p$ (which can be done thanks to Tonelli's and Shanks' algorithm), and then to apply Euclid's algorithm to (p, x_0) until we obtain a remainder $r < \sqrt{p}$. One can then prove that if $c = (p - r^2)/d$ is the square of an integer, say s^2, then $(x, y) = (r, s)$ is a solution, and that otherwise there is no solution. Many proofs of this result can be found in the literature.

7) Use Cornacchia's algorithm to solve $x^2 + 2y^2 = 97$.

8) Admit that there is at least one solution. Does Cornacchia's algorithm allow to find all the solutions?

9) Evaluate the algebraic and word complexities of the second part of the algorithm.