Université de Bordeaux
Mathématiques

N1MA9W05 – Master CSI
2015–2016

Partial Exam, October 19th 2015 (2pm – 4pm)
Duration 2 hours. All documents allowed.

Clarity of programs and comments is a major factor in the rating scale.

- To answer the questions, create *a single* file per exercise, named `login1.gp`, then `login2.gp`, etc. (Type `whoami` in a terminal if you are unsure about your login.) For instance, `kbelabas1.gp`.
- To hand over your answers, type `~kbelabas/copy` in a terminal, from the directory where your files are saved. (You may do this multiple times, only the last one matters : copies made previously are replaced.)

Exercise 1 – Find a primality certificate for the integer $300 \times 2^{2015} + 1$.

Exercise 2 –

1) Eratosthenes's basic sieve computes primes $\leqslant B$ via an array T of length B such that $T[i] = 1$ if and only if i is prime. For all consecutive primes p the sieve sets $T[i]$ to 0 in a loop of type `forstep(i = p^2, B, p, T[i] = 0)`. Program such a sieve.

2) Taking care not to forget the prime 2, it is useless to consider even numbers. Write a new program yielding an array T of length $\approx B/2$ such that $T[i] = 1$ if and only if $2i + 1$ is prime. Note that if $p \geqslant 3$ is prime, then p^2 is odd and the $i = p^2 + p$, $p^2 + 3p$, $p^2 + 5p$, etc. are even, hence pointless. How much can we hope to gain compared to 1) ?

3) Let's go further : fix δ invertible mod $30 = 2 \times 3 \times 5$.

a) Obtain the list of all primes of the form $30i + \delta$, via an array T (of length $\approx B/30$) such that $T[i] = 1$ if and only if $30i + \delta$ is prime. How much can we hope to gain ?

b) Obtain the list of all allowed $\delta \in (\mathbb{Z}/30\mathbb{Z})^*$.

★ **4)** Can one generalize further and continue to gain ?

Around the Fast Fourier Transform (FFT).

Let $n > 1$ be an integer and let K be a commutative field containing a primitive n-th root of unity ω. In other words, ω has order exactly n in $(K^*, \times)$. We define $\omega^0 = 1$. If $K = \mathbb{F}_q$ is a finite field, such an ω exists if and only if $n \mid (q - 1)$.

Exercise 3 – [Examples]

1) Prove that the characteristic of K can never divide n.

2) Find such an ω for $q =$ `nextprime(10^60)` and $n = q - 1$.

3) Find such an ω of multiplicative order 2^{16} in a quadratic finite field $\mathbb{F}_{p^2}$, such that $\omega \notin \mathbb{F}_p$.

4) Fix $n = 2^{32}$; find a prime p such that $n \mid p - 1$, then an ω of order n in $\mathbb{F}_p^*$.

Exercise 4 – [Fourier transform]
Let $(a_i\colon 0 \leqslant i < n) \in K^n$; by abuse of notation, we identify such a vector with the polynomial $f = \sum_{0 \leqslant i<n} a_i X^i$ in $K[X]_{<n}$. The Fourier Transform of (a_i) relatively to ω is the vector

$$\mathcal{F}((a_i), \omega) = \mathcal{F}(f, \omega) := (b_j\colon 0 \leqslant j < n) \in K^n, \quad \text{where} \quad b_j = f(\omega^j).$$

If $k \in \mathbb{Z}$, we have $\sum_{0 \leqslant i<n} \omega^{ik} = 0$ if $n \nmid k$, and that sum is n otherwise. It follows that

$$\mathcal{F}((b_j\colon 0 \leqslant j < n), \omega^{-1}) = (na_i\colon 0 \leqslant i < n),$$

which yields a simple formula for the inverse transform $\mathcal{F}^{-1}(\cdot, \omega) = \frac{1}{n}\mathcal{F}(\cdot, \omega^{-1})$.

1) Program a naive algorithm to compute the Fourier transform of (a_i) using $O(n^2)$ operations in K (additions and multiplications).

2) Same question for the inverse transform.

Exercise 5 – [FFT]
We assume from now on that $n = 2^k$, for some integer $k \geqslant 1$. Let $f \in K[X]_{<n}$, whose degree is less than n, we define f_{even} and f_{odd} by

$$f = \sum_{i=0}^{n-1} a_i X^i = f_{\text{even}}(X^2) + X \cdot f_{\text{odd}}(X^2).$$

Let

$$(u_i\colon 0 \leqslant i < n/2) := \mathcal{F}(f_{\text{even}}, \omega^2),$$
$$(v_i\colon 0 \leqslant i < n/2) := \mathcal{F}(f_{\text{odd}}, \omega^2).$$

By abuse of language, we extend u_j and v_j to $j \in \mathbb{Z}$ by periodicity modulo $n/2$. We then have $f(\omega^j) = u_j + \omega^j v_j$ for all $j \in \mathbb{Z}$.

1) Program a recursive algorithm for $\mathcal{F}(f, \omega)$ using the previous formulae.

2) If your original program did not do so, write a new version assuming that the vector of all ω^i, $i < n$, are precomputed.

3) The FFT algorithm uses $O(n \log n)$ opérations in K. For a few well-chosen fields, determine experimental thresholds where the recursive algorithm beats the naive one.