CS60094 Computational Number Theory

Mid-Semester Test

Maximum marks: 30 February 26, 2010 **Duration:** 2 hours

Roll No | | | | | | | | **Name**

[*This test is open-notes. Answer all questions. Be brief and precise.*]

1 Suppose that $\gcd(r_0, r_1)$ is computed by the repeated Euclidean division algorithm. Suppose also that $r_0 > r_1 > 0$. Let r_{i+1} denote the remainder obtained by the i-th division (that is, in the i-th iteration of the Euclidean loop). So the computation proceeds as $\gcd(r_0, r_1) = \gcd(r_1, r_2) = \gcd(r_2, r_3) = \cdots$ with $r_0 > r_1 > r_2 > \cdots > r_k > r_{k+1} = 0$ for some $k \geqslant 1$.

(a) If the computation of $\gcd(r_0, r_1)$ requires exactly k Euclidean divisions, show that $r_0 \geqslant F_{k+2}$ and $r_1 \geqslant F_{k+1}$. Here, F_n is the n-th Fibonacci number: $F_0 = 0$, $F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$ for $n \geqslant 2$. **(4)**

Solution For each i in the range $2 \leqslant i \leqslant k+1$, we have $r_{i-2} = q_i r_{i-1} + r_i$. Since $r_{i-2} > r_{i-1}$, we have $q_i \geqslant 1$, so $r_{i-2} \geqslant r_{i-1} + r_i$. Moreover, $r_k \neq 0$, that is, $r_k \geqslant 1 = F_2$, and $r_{k-1} > r_k$, that is, $r_{k-1} \geqslant 2 = F_3$. But then $r_{k-2} \geqslant r_{k-1} + r_k \geqslant F_3 + F_2 = F_4$, $r_{k-3} \geqslant r_{k-2} + r_{k-1} \geqslant F_4 + F_3 = F_5$, and so on. Proceeding in this way, we can show that $r_1 \geqslant F_{k+1}$, and $r_0 \geqslant F_{k+2}$.

(b) Modify the Euclidean gcd algorithm slightly so as to ensure that $r_i \leqslant \frac{1}{2} r_{i-1}$ for $i \geqslant 2$. Here, r_i need not be the remainder $r_{i-2} \operatorname{rem} r_{i-1}$. **(4)**

Solution Compute $r_i = r_{i-2} \operatorname{rem} r_{i-1}$, where $0 \leqslant r_i \leqslant r_{i-1} - 1$. If $r_i > \frac{1}{2} r_{i-1}$, replace r_i by $r_{i-1} - r_i$. The correctness of this variant is based on the fact that $\gcd(r_{i-1}, r_i) = \gcd(r_{i-1}, -r_i) = \gcd(r_{i-1}, r_{i-1} - r_i)$.

(c) Explain the speedup produced by the modified algorithm. You may assume that $F_n \approx \frac{1}{\sqrt{5}}\rho^n$, where $\rho = \frac{1+\sqrt{5}}{2} = 1.6180339887\ldots$ is the golden ratio. **(4)**

Solution In the original Euclidean algorithm, we have $r_1 \geqslant F_{k+1} \approx \frac{1}{\sqrt{5}}\rho^{k+1}$, that is, $k \leqslant -1 + (\log \sqrt{5} r_1)/\log \rho$. For the modified algorithm, let k' denote the number of iterations. We have $r_1 \geqslant 2r_2 \geqslant 2^2 r_3 \geqslant \cdots \geqslant 2^{k'-1} r_{k'} \geqslant 2^{k'-1}$, that is, $k' \leqslant 1 + \log(r_1)/\log(2)$. Since $2 > \rho$, the modified algorithm has the potential of reducing the number of iterations of the Euclidean loop by a factor of $\log 2/\log \rho$.

2 Represent $\mathbb{F}_{64} = \mathbb{F}_{2^6}$ as $\mathbb{F}_2(\theta)$ with $\theta^6 + \theta^3 + 1 = 0$.

(a) Find all the conjugates of θ (over $\mathbb{F}_2$ as polynomials in θ of degrees < 6). **(4)**

Solution The conjugates of θ are

$$\begin{aligned}
&\theta, \\
&\theta^2, \\
&\theta^4, \\
&\theta^8 = \theta^2(\theta^3+1) = \theta^5 + \theta^2, \\
&\theta^{16} = \theta^{10} + \theta^4 = \theta^4(\theta^3+1) + \theta^4 = \theta^7 = \theta^4 + \theta, \text{ and} \\
&\theta^{32} = \theta^8 + \theta^2 = \theta^2(\theta^3+1) + \theta^2 = \theta^5.
\end{aligned}$$

(b) Prove or disprove: θ is a primitive element of $\mathbb{F}_{64}^*$. **(4)**

Solution It suffices to compute θ^h only for $h|63$. Now, $\theta \neq 1$, $\theta^3 \neq 1$, $\theta^7 = \theta(\theta^3+1) = \theta^4 + \theta \neq 1$, and $\theta^9 = \theta^3(\theta^3+1) = \theta^6 + \theta^3 = 1$. That is, the order of θ is 9, that is, θ is not a primitive element of $\mathbb{F}_{64}^*$.

Alternatively, by Part (a), $\theta^{32} = \theta^5$, that is, $\theta^{27} = 1$, that is, $\operatorname{ord}\theta$ divides 27 and so is smaller that $64 - 1 = 63$.

(c) What is the minimal polynomial of θ^3 over $\mathbb{F}_2$? **(4)**

Solution We have $\theta^6 + \theta^3 + 1 = 0$, that is, $(\theta^3)^2 + (\theta^3) + 1 = 0$, that is, $f_{\theta^3,2}(x) = x^2 + x + 1$.

If you choose, you may go as computers would do, that is, write $\alpha = \theta^3$, then show that $\alpha^2 = \theta^3 + 1$ and $\alpha^4 = \theta^6 + 1 = \theta^3 = \alpha$, so that $f_{\theta^3,2}(x) = (x-\alpha)(x-\alpha^2) = (x+\theta^3)(x+\theta^3+1) = x^2 + x + 1$.

3 Let p be a prime congruent to 3 modulo 4, and a the last four digits of your roll number. You may assume that $p \nmid a$. Prove that the congruence $y^2 \equiv x^3 + ax \pmod{p}$ has exactly p solutions for (x, y) modulo p. **(8)**

Solution Guess what! The result does not depend upon your roll number. So I work with any arbitrary integer a with $p \nmid a$. Indeed, the condition $p \nmid a$ is also not necessary for this exercise, but e-mails require this condition.

If $x \equiv 0 \pmod{p}$, the only solution for y in the given congruence is 0.

So consider $x \not\equiv 0 \pmod{p}$. In this case, x and $-x$ are distinct modulo p. I will now show that for the pair of values $\pm x$, we have exactly two solutions of the given congruence. If $x^2 + a \equiv 0 \pmod{p}$, these two solutions are $(x, 0)$ and $(-x, 0)$. If not, we look at the Legendre symbols $\left(\frac{x^3+ax}{p}\right) = \left(\frac{x}{p}\right)\left(\frac{x^2+a}{p}\right)$ and $\left(\frac{(-x)^3+a(-x)}{p}\right) = \left(\frac{-1}{p}\right)\left(\frac{x}{p}\right)\left(\frac{x^2+a}{p}\right) = -\left(\frac{x^3+ax}{p}\right)$, where the last equality follows from the fact that $\left(\frac{-1}{p}\right) = -1$ for a prime $p \equiv 3 \pmod{4}$. It then follows that exactly one of $\pm x$ yields exactly two solutions (for y) of the given congruence, whereas the other leads to no solutions (for y).

That's all!

(**Remark:** Because of its usability in e-mails, this is an important congruence for computer engineers. Indeed, it is this number of solutions, that is the source of all its importance :–)