CS60082 Computational Number Theory, Spring 2008

End-semester examination: Solutions

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

1 Let n be an odd composite integer and $\gcd(a,n) = 1$. Prove the following assertions.

(a) If n is an Euler pseudoprime to base a, then n is a (Fermat) pseudoprime to base a. **(5)**

Solution Let n be an Euler pseudoprime to base a. Then $a^{(n-1)/2} \equiv \left(\frac{a}{n}\right) \pmod{n}$. Since $\left(\frac{a}{n}\right) = \pm 1$, squaring gives $a^{n-1} \equiv 1 \pmod{n}$, that is, n is a pseudoprime to base a.

(b) There exists a base to which n is not an Euler pseudoprime. **(15)**

Solution In view of Part (a), it suffices to concentrate only on Carmichael numbers n. We can write $n = p_1 p_2 \cdots p_r$ with pairwise distinct odd primes $p_1, p_2, \ldots, p_r$, $r \geqslant 3$, and with $(p_i - 1) \mid (n-1)$ for all $i = 1, 2, \ldots, r$. We now consider two cases.

Case 1: All $\frac{n-1}{p_i-1}$ are even.

We choose a base $a \in \mathbb{Z}_n^*$ such that $\left(\frac{a}{p_1}\right) = -1$, whereas $\left(\frac{a}{p_i}\right) = +1$ for $i = 2, 3, \ldots, r$. By the definition of the Jacobi symbol, we have $\left(\frac{a}{n}\right) = -1$. By Euler's criterion, $a^{(p_1-1)/2} \equiv -1 \pmod{p_1}$. Since $\frac{n-1}{p_1-1} = \frac{(n-1)/2}{(p_1-2)/2}$ is even by hypothesis, we have $a^{(n-1)/2} \equiv 1 \pmod{p_1}$. On the other hand, for $i = 2, 3, \ldots, r$, we have $a^{(p_i-1)/2} \equiv 1 \pmod{p_i}$, that is, $a^{(n-1)/2} \equiv 1 \pmod{p_i}$. By CRT, we then have $a^{(n-1)/2} \equiv 1 \pmod{n}$, that is, $a^{(n-1)/2} \not\equiv \left(\frac{a}{n}\right) \pmod{n}$, that is, n is not an Euler pseudoprime to base a.

Case 2: Some $\frac{n-1}{p_i-1}$ is odd.

Without loss of generality, assume that $\frac{n-1}{p_1-1}$ is odd. Again take $a \in \mathbb{Z}_n^*$ with $\left(\frac{a}{p_1}\right) = -1$ and $\left(\frac{a}{p_i}\right) = +1$ for $i = 2, 3, \ldots, r$. By the definition of the Jacobi symbol, we then have $\left(\frac{a}{n}\right) = -1$. On the other hand, by Euler's criterion, we have $a^{(n-1)/2} \equiv -1 \pmod{p_1}$ and $a^{(n-1)/2} \equiv 1 \pmod{p_i}$ for $i = 2, 3, \ldots, r$. By CRT, we conclude that $a^{(n-1)/2} \not\equiv \pm 1 \pmod{n}$, that is, $a^{(n-1)/2} \not\equiv \left(\frac{a}{n}\right) \pmod{n}$, that is, n is not an Euler pseudoprime to base a.

(c) n is an Euler pseudoprime to at most half the bases in $\mathbb{Z}_n^*$. **(5)**

Solution Suppose that n is an Euler pseudoprime to the bases $a_1, a_2, \ldots, a_t \in \mathbb{Z}_n^*$ only. Let a be a base to which n is not an Euler pseudoprime. (Such a base exists by Part (b).) We have $a^{(n-1)/2} \not\equiv \left(\frac{a}{n}\right) \pmod{n}$. On the other hand, $a_i^{(n-1)/2} \equiv \left(\frac{a_i}{n}\right) \pmod{n}$ for $i = 1, 2, \ldots, t$. It follows that $(aa_i)^{(n-1)/2} \equiv a^{(n-1)/2} a_i^{(n-1)/2} \not\equiv \left(\frac{a}{n}\right)\left(\frac{a_i}{n}\right) \equiv \left(\frac{aa_i}{n}\right) \pmod{n}$, that is, n is not an Euler pseudoprime to each of the bases aa_i, that is, there are at least t bases to which n is not an Euler pseudoprime.

2 The *Lehmer sequence* with parameters a, b is defined as

$$
\begin{aligned}
\bar{U}_0 &= 0, \\
\bar{U}_1 &= 1, \\
\bar{U}_m &= \bar{U}_{m-1} - b\bar{U}_{m-2} \text{ if } m \geqslant 2 \text{ is even}, \\
\bar{U}_m &= a\bar{U}_{m-1} - b\bar{U}_{m-2} \text{ if } m \geqslant 3 \text{ is odd}.
\end{aligned}
$$

Let α, β be the roots of $x^2 - \sqrt{a}x + b$.

(a) Prove that $\bar{U}_m = \begin{cases} (\alpha^m - \beta^m)/(\alpha^2 - \beta^2) & \text{if } m \text{ is even}, \\ (\alpha^m - \beta^m)/(\alpha - \beta) & \text{if } m \text{ is odd}. \end{cases}$ **(10)**

Solution We proceed by induction on m. For $m = 0$, we have $\bar{U}_0 = (\alpha^0 - \beta^0)/(\alpha^2 - \beta^2) = 0$, whereas for $m = 1$, we have $\bar{U}_1 = (\alpha^1 - \beta^1)/(\alpha - \beta) = 1$. Now suppose that $\bar{U}_{2k} = (\alpha^{2k} - \beta^{2k})/(\alpha^2 - \beta^2)$ and $\bar{U}_{2k+1} = (\alpha^{2k+1} - \beta^{2k+1})/(\alpha - \beta)$ for some $k \geqslant 0$. Since α, β are roots of $x^2 - \sqrt{a}x + b$, we have $\alpha + \beta = \sqrt{a}$ and $\alpha\beta = b$. But then

$$\begin{aligned}
\bar{U}_{2k+2} &= \bar{U}_{2k+1} - b\bar{U}_{2k} \\
&= \left(\frac{\alpha^{2k+1} - \beta^{2k+1}}{\alpha - \beta}\right) - b\left(\frac{\alpha^{2k} - \beta^{2k}}{\alpha^2 - \beta^2}\right) \\
&= \frac{(\alpha + \beta)(\alpha^{2k+1} - \beta^{2k+1}) - b(\alpha^{2k} - \beta^{2k})}{\alpha^2 - \beta^2} \\
&= \frac{(\alpha^{2k+2} - \beta^{2k+2}) + (\alpha^{2k} - \beta^{2k})(\alpha\beta - b)}{\alpha^2 - \beta^2} \\
&= \frac{\alpha^{2k+2} - \beta^{2k+2}}{\alpha^2 - \beta^2}.
\end{aligned}$$

On the other hand,

$$\begin{aligned}
\bar{U}_{2k+3} &= a\bar{U}_{2k+2} - b\bar{U}_{2k+1} \\
&= a\left(\frac{\alpha^{2k+2} - \beta^{2k+2}}{\alpha^2 - \beta^2}\right) - b\left(\frac{\alpha^{2k+1} - \beta^{2k+1}}{\alpha - \beta}\right) \\
&= \frac{a(\alpha^{2k+2} - \beta^{2k+2}) - b(\alpha + \beta)(\alpha^{2k+1} - \beta^{2k+1})}{\alpha^2 - \beta^2} \\
&= \frac{(\alpha + \beta)^2(\alpha^{2k+2} - \beta^{2k+2}) - \alpha\beta(\alpha + \beta)(\alpha^{2k+1} - \beta^{2k+1})}{\alpha^2 - \beta^2} \\
&= \frac{(\alpha + \beta)(\alpha^{2k+2} - \beta^{2k+2}) - \alpha\beta(\alpha^{2k+1} - \beta^{2k+1})}{\alpha - \beta} \\
&= \frac{\alpha^{2k+3} - \beta^{2k+3}}{\alpha - \beta}.
\end{aligned}$$

(b) Let $\Delta = a - 4b$ and n a positive integer with $\gcd(n, 2a\Delta) = 1$. We call n is *Lehmer pseudoprime* with parameters a, b if $\bar{U}_{n-\left(\frac{a\Delta}{n}\right)} \equiv 0 \pmod{n}$. Prove that n is a Lehmer pseudoprime with parameters a, b if and only if n is a Lucas pseudoprime with parameters a, ab. **(10)**

Solution For the Lehmer sequence with parameters a, b, we take $\alpha = \frac{\sqrt{a}+\sqrt{a-4b}}{2}$ and $\beta = \frac{\sqrt{a}-\sqrt{a-4b}}{2}$. The discriminant is $\Delta = a - 4b$. On the other hand, for the Lucas sequence with parameters a, ab, the roots of the characteristic equation are $\alpha' = \frac{a+\sqrt{a^2-4ab}}{2}$ and $\beta' = \frac{a-\sqrt{a^2-4ab}}{2}$. Also the discriminant is $\Delta' = a^2 - 4ab$. That is, $\alpha' = \sqrt{a}\alpha$, $\beta' = \sqrt{a}\beta$ and $\Delta' = a\Delta$. Finally, note that $n - \left(\frac{a\Delta}{n}\right)$ is even. Therefore,

$$\begin{aligned}
\bar{U}_{n-\left(\frac{a\Delta}{n}\right)} &= \frac{\alpha^{n-\left(\frac{a\Delta}{n}\right)} - \beta^{n-\left(\frac{a\Delta}{n}\right)}}{\alpha^2 - \beta^2} \\
&= \frac{\alpha^{n-\left(\frac{\Delta'}{n}\right)} - \beta^{n-\left(\frac{\Delta'}{n}\right)}}{(\alpha + \beta)(\alpha - \beta)} \\
&= \frac{(\sqrt{a})^{-\left[n-\left(\frac{\Delta'}{n}\right)\right]}\left(\alpha'^{n-\left(\frac{\Delta'}{n}\right)} - \beta'^{n-\left(\frac{\Delta'}{n}\right)}\right)}{\sqrt{a}(\alpha - \beta)} \\
&= \frac{1}{(\sqrt{a})^{n-\left(\frac{\Delta'}{n}\right)}}\left(\frac{\alpha'^{n-\left(\frac{\Delta'}{n}\right)} - \beta'^{n-\left(\frac{\Delta'}{n}\right)}}{\alpha' - \beta'}\right) \\
&= \frac{U'_{n-\left(\frac{\Delta'}{n}\right)}}{(\sqrt{a})^{n-\left(\frac{\Delta'}{n}\right)}},
\end{aligned}$$

where U'_m is the m-th term in the corresponding Lucas sequence. Since $\gcd(a, n) = 1$, it follows that $\bar{U}_{n-\left(\frac{a\Delta}{n}\right)} \equiv 0 \pmod{n}$ if and only if $U'_{n-\left(\frac{\Delta'}{n}\right)} \equiv 0 \pmod{n}$.

3 Prove that for $m \geqslant 2$, the Fermat number $f_m = 2^{2^m} + 1$ is prime if and only if $5^{(f_m-1)/2} \equiv -1 \pmod{f_m}$. **(10)**

Solution The condition $5^{(f_m-1)/2} \equiv -1 \pmod{f_m}$ implies that $\operatorname{ord}_{f_m}(5) = f_m - 1$, that is, f_m is prime. Conversely, suppose that f_m is prime. By Euler's criterion, $5^{(f_m-1)/2} \equiv \left(\frac{5}{f_m}\right) \pmod{f_m}$. But by the quadratic reciprocity law, $\left(\frac{5}{f_m}\right) = (-1)^{(f_m-1)(5-1)/4}\left(\frac{f_m}{5}\right) = \left(\frac{f_m}{5}\right) = \left(\frac{2^{2^m}+1}{5}\right) = \left(\frac{4^{2^{m-1}}+1}{5}\right) = \left(\frac{(-1)^{2^{m-1}}+1}{5}\right) = \left(\frac{1+1}{5}\right) = \left(\frac{2}{5}\right) = -1$.

4 **(a)** Suppose you are given a black-box that, given two positive integers n and k, returns in one unit of time the decision whether n has a factor d in the range $2 \leqslant d \leqslant k$. Using this black-box, devise an algorithm to factor a positive integer n in polynomial (in $\log n$) time. **(10)**

Solution We implement a binary search procedure for locating a non-trivial factor of n. The steps are listed below. We maintain two bounds L, U with $L \leqslant U$.

```
If the black-box returns 'no' for input n, n−1, return 'n is prime'.
Set L = 2 and U = n − 1.
while (L < U) {
    Set M = (L + U)/2.
    If the black-box returns 'yes' for input n, M, set U = M,
    else set L = M + 1.
}
return L.
```

(b) Deduce the running time of your algorithm. **(5)**

Solution The `while` loop runs for $O(\log n)$ times. Each iteration of the loop takes $O(\log n)$ time. Thus the running time of our algorithm is $O(\log^2 n)$.

5 Write a pseudocode implementing Floyd's variant of Pollard's rho method with block gcd calculations. **(10)**

Solution Suppose that we use a block of t gcd's.

```
Initialize x and y to a random element of Z_n.
Also initialize a running product p = 1 and a running count k = 0.
Finally, initialize values of x, y before the current block: x' = x and y' = y.
while (1) {
    Update x = f(x) and y = f(f(y)).
    Update the product p ≡ p × (x − y) (mod n) and the count k = k + 1.
    if k equals t {
        Compute the gcd d = gcd(p, n).
        if d equals 1 {
            Prepare for the next block: p = 1, k = 0, x' = x and y' = y.
        } else {
            Go back to the start of the current block: x = x' and y = y'.
            while (1) {
                Recalculate x = f(x) and y = f(f(y)).
                Compute individual gcd d = gcd(n, x − y).
                if (d > 1) return d.
            }
        }
    }
}
```

6 **(a)** Explain how sieving is carried out in connection with the multiple-polynomial quadratic sieve method, that is, for the general polynomial $T(c) = U + 2Vc + Wc^2$ with $V^2 - UW = n$. **(10)**

Solution We initialize an array A indexed in the range $-M \leqslant c \leqslant M$. The array location A_c is initialized to $\log|T(c)|$.

Let p be a small prime in the factor base. If $p = 2$, we obtain the multiplicity m_c of 2 in $T(c)$ by bit operations. We subtract $m_c \log 2$ from A_c. (The array location A_c may be initialized after all factors of 2 are extracted from $T(c)$.)

Now let p be an odd prime and h a small exponent. We have $WT(c) = (Wc+V)^2 - n$, so the condition $p^h \mid T(c)$ is equivalent to $(Wc+V)^2 \equiv n \pmod{p^h}$. Since n is a quadratic residue modulo p, this congruence has exactly two solutions for c. For $h = 1$, these solutions are obtained by a root finding algorithm in $\mathbb{Z}_p$, whereas for $h > 1$, the solutions are obtained by Hensel lifting. Let c_1, c_2 be the two solutions. For each c in the range $-M \leqslant c \leqslant M$ with $c \equiv c_1, c_2 \pmod{p^h}$, we subtract $\log p$ from the array location A_c.

After all small primes p in the factor base are considered, we look at the values left in A_c. If $A_c \approx 0$ for some c, we factor $T(c)$ by trial division by factor base primes and obtain a relation.

(b) Assume that the factor base consists of $L[1/2]$ primes and the sieving interval is of size $L[1]$. Deduce that the sieving process can be completed in $L[1]$ time. **(10)**

Solution Each value of $T(c)$ and its (approximate) logarithm can be computed in time polynomial in $\log n$. Thus the array A can be initialized in $L[1]$ time.

The multiplicity m_c of the prime $p = 2$ in each $T(c)$ can be obtained in $O(\log n)$ time and subsequently a suitable right shift operation removes the factors of 2 from each $T(c)$ again in $O(\log n)$ time. Since there are $2M + 1 = L[1]$ values of $T(c)$ to consider, this step takes $L[1]$ time.

For each small prime power p^h, one first obtains the two solutions c_1, c_2. This is doable in (probabilistic) polynomial time. Subsequently, one updates appropriate locations A_c. For a given p^h, the total time for subtraction of $\log p$ from all appropriate locations is $\approx (2M+1)/p^h$. Summing over all values of p, h gives a total running time of $O(\log n)M$ which is $L[1]$.

Finally, we scan over the entire array A in $2M + 1 = L[1]$ time. We expect $L[1/2]$ relations. For each such relation, trial division by $L[1/2]$ primes in the factor base requires $L[1/2]$ time. Thus, the time for factoring all smooth values of $T(c)$ is $L[1/2] \times L[1/2] = L[1]$.