Indian Institute of Technology Kharagpur
Department of Computer Science and Engineering

CS60094 Computational Number Theory **Spring 2026**

Mid-Semester Examination **19 February, 3 PM – 5 PM** **Marks = 60**

*Answer **any four** questions. State all assumptions you make. Keep your answers concise.*

1. Consider the following algorithm which takes as input $a, b \in \mathbb{N}$.

   B-Euclid(a, b)
   set $r \leftarrow a$, $s \leftarrow b$, $k \leftarrow 0$
   while $r \equiv 0 \pmod 2$ and $s \equiv 0 \pmod 2$ do
   $r \leftarrow r/2$, $s \leftarrow s/2$, $k \leftarrow k + 1$
   while $s > 0$ do
   while $r \equiv 0 \pmod 2$ do $r \leftarrow r/2$
   while $s \equiv 0 \pmod 2$ do $s \leftarrow s/2$
   if $s < r$, set $(r, s) \leftarrow (s, r)$
   $s \leftarrow s - r$
   return $2^k \cdot r$

   Prove that

   (a) B-Euclid(a, b) computes and returns $\gcd(a, b)$.

   (b) running time of the algorithm is $O(n^2)$ where $n = \max\{\log a, \log b\}$. (Observe that division by 2 is nothing but a right bit-shift operation.)

   10+5 = 15

2. Let $m \in \mathbb{N}$ be a large modulus and let $a, b \in \mathbb{Z}_m$. Assume $(a - 1) \in \mathbb{Z}_m^*$. Consider a sequence of integers $x_1, x_2, x_3, \ldots$ modulo m, computed by first choosing x_0 from $\mathbb{Z}_m$ and recursively generating $x_i \leftarrow ax_{i-1} + b \pmod m$ for $i \geq 1$.

   (a) Describe an algorithm to compute x_n given (x_0, a, b, n) that runs in time polynomial in both $\log_2 m$ and $\log_2 n$.

   (b) What is the running time of your algorithm?

   12+3 = 15

3. (a) Let m_1, m_2 be coprime moduli and let $a_1, a_2 \in \mathbb{Z}$. By the extended gcd algorithm, one can compute $u, v \in \mathbb{Z}$ such that $um_1 + vm_2 = 1$. Prove that $x = um_1a_2 + vm_2a_1 \pmod{m_1m_2}$ is the simultaneous solution of the congruences $x \equiv a_i \pmod{m_i}$ for $i = 1, 2$.

   (b) Let $m_1, m_2, \ldots, m_t$ be pairwise coprime moduli, and $a_1, a_2, \ldots, a_t \in \mathbb{Z}$. Write an *incremental procedure* for the Chinese remainder theorem that starts with the solution $x \equiv a_1 \pmod{m_1}$ and then runs a loop (for $i = 2, 3, \ldots, t$ in that order), the i-th iteration of which computes the simultaneous solution of $x \equiv a_j \pmod{m_j}$ for $j = 1, 2, \ldots, i$. Prove the correctness of your algorithm.

   5+10 = 15

4. Let p be an odd prime and $e \in N$.
   (a) For $a, b \in \mathbb{Z}_{p^e}^*$, show that $a^2 \equiv b^2 \pmod{p^e}$ if and only if $a \equiv \pm b \pmod{p^e}$.
   (b) Prove that $|\mathsf{QR}_{p^e}| = \phi(p^e)/2$.
   (Here, QR_m denotes the set of quadratic residues modulo m).

$\boxed{6+9 = 15}$

5. Let m be a modulus having a primitive root, and $a \in \mathbb{Z}_m^*$.
   (a) Prove that a is a primitive root modulo m if and only if $a^{\phi(m)/q} \not\equiv 1 \pmod{m}$ for every prime divisor q of $\phi(m)$.
   (b) Design an algorithm that, given $a \in \mathbb{Z}_m^*$ and the prime factorization of $\phi(m)$, determines whether a is a primitive root modulo m.

$\boxed{7+8 = 15}$