Indian Institute of Technology Kharagpur
Department of Computer Science and Engineering

CS60094 Computational Number Theory **Spring 2026**

End Semester Examination **19 April, 3 PM – 5 PM** **Marks = 60**

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

1. Let $p \in \mathbb{P}$. Prove or Disprove:
   (a) Let $a \in \mathbb{Z}_p^*$. The polynomial $f(x) = x^p - x - a \in \mathbb{Z}_p[x]$ is irreducible.
   (b) Let $\alpha \in \mathbb{F}_{p^n}^*$. The elements $\alpha, \alpha^p, \alpha^{p^2}, \ldots, \alpha^{p^{n-1}}$ (conjugates of α) do not necessarily have the same order.

   6+6 = 12

2. Prove the following statements.
   (a) Let $n \in \mathbb{N}$, $n > 1$. There exists an element in $\mathbb{Z}_n^*$ of multiplicative order $n-1$ if and only if n is a prime.
   (b) Let p be a prime. $n = 2p + 1$ is a prime if and only if $2^{n-1} \equiv 1 \bmod n$.

   6+6 = 12

3. Assume the following: *a composite integer $n > 1$ is a Carmichael number if and only if (i) n is squarefree and (ii) for every prime p dividing n, we have $(p-1)|(n-1)$.*

   Let n be a Carmichael number. Prove/Disprove the following.
   (a) n is odd
   (b) n has atleast 3 distinct prime factors.
   (c) $\gcd(n, \phi(n)) > 1$.

   4+4+4 = 12

4. An odd number of the form $2^r k + 1$ where $r \geq 1$, k odd and $2^r > k$ is called a *Proth number*. A Proth number that is prime is called a *Proth prime*.
   (a) Describe an efficient way to determine whether a given odd positive integer is a Proth number.
   (b) Suppose that a Proth number $n = 2^r k + 1$ satisfies $a^{(n-1)/2} \equiv -1 \pmod{n}$ for some integer $a > 1$. Prove that n is prime.
   (c) Devise a yes-biased probabilistic polynomial-time algorithm to test the primality of a Proth number.
   (d) Analyse the probability of error (that is, the probability that a prime is falsely identfied as composite).

   2+4+3+3 = 12

5. Dixon's method for factoring an integer n can be combined with a sieve in order to reduce its running time from $L[2]$ to $L[3/2]$. Instead of choosing random values of $x_1, x_2, \ldots, x_s$ in the relations, we first choose a random value of x, and for $-M \leq c \leq M$, we check the smoothness of the integers $(x+c)^2 \bmod n$ over t small primes $p_1, p_2, \ldots, p_t$. As in Dixon's original method, we take $t = L[1/2]$.
   (a) Determine M for which one expects to get a system of the desired size.
   (b) Describe a sieve over the interval $[-M, M]$ for detecting the smooth values of $(x+c)^2 \bmod n$.
   (c) Deduce how you achieve a running time of $L[3/2]$ using this sieve.

   3+6+3 = 12