CS681 - COMPUTATIONAL NUMBER THEORY & ALGEBRA

NITIN SAXENA

END-SEMESTER EXAMINATION

POINTS: 65

GIVEN: 30-APR-2025 DUE: 03-MAY-2025 (5PM)

Rules:

- Solve it independently. You are *not* allowed to discuss. You are *not* allowed to copy the language of chatGPT or other generative-AI tools.
- Write the solutions on your own and honorably *acknowledge* the sources if any. `cse.iitk.ac.in/pages/AntiCheatingPolicy.html`
- Clearly express the fundamental *idea* of your proof/ algorithm before going into the other proof details. The distribution of partial marks is according to the proof steps.
- There will be a penalty if you write unnecessary or unrelated details in your solution. Also, do not repeat the proofs done in the class or the assignments.
- Submit your solutions, before time, to your TA (CC: Instructor). Preferably, submit a printed/pdf copy of your LaTeXed or Word processed solution sheet.

  Your TA will help in the doubt resolution: `Tufan Singha Mahapatra <tufansm@cse.iitk.ac.in>`

Question 1: *[5+10 points]* Can $n = pq$, for distinct primes p and q, be a Carmichael number?

- Design an efficient algorithm to find a nontrivial factor of a given Carmichael number.

Solve only one out of the Questions 2.1 and 2.2.

Question 2.1: *[20 points]* Let $f_1, f_2 \in \mathbb{F}_q[X]$ be two univariate polynomials over a finite field. Consider the algebras $A_i := \mathbb{F}_q[X]/\langle f_i \rangle$, $i \in [2]$.

Design an efficient algorithm to find an (algebra) isomorphism between the two algebras, or decide their non-isomorphism.

Solve only one out of the Questions 2.1 and 2.2.

Question 2.2: *[8+12 points]* Let n be a given integer. For an $a \in \mathbb{Z}$ consider the ring $R_a := (\mathbb{Z}/n\mathbb{Z})[X]/\langle X^2 - a\rangle$. Show that

(1) An efficient algorithm to *count* the (ring) automorphisms of R_0 implies the existence of an efficient way to factor n (hence, breaking the RSA!).
(2) An efficient algorithm to *find* a (ring) isomorphism from R_1 to R_a, for any input a, implies the existence of an efficient way to factor n.

Question 3: *[20+10 points]* Let $f(x, y)$ be a bivariate polynomial over $\mathbb{Z}/n\mathbb{Z}$ with the degrees of variables x and y bounded by ℓ and m respectively, with $f(x, y) = x^\ell y^m +$ lower degree terms. Define the set of "small" roots of f modulo n, i.e.

$$P_\epsilon := \left\{(a, b) \in \mathbb{Z}^2 \mid |a|, |b| \leq n^\epsilon,\ f(a, b) \equiv 0 \,(\text{mod } n)\right\} .$$

- Design an efficient algorithm that, for $\epsilon \leq \frac{1}{2(\ell+m+2)}$ and for large enough n, finds a nonzero polynomial $h(x, y)$ such that if $(a, b) \in P_\epsilon$ then $h(a, b) = 0$ in $\mathbb{Z}$.
- Design an efficient algorithm that, for the above bound on ϵ, computes all degree d curves $y = g(x)$ such that

$$\#\left\{(a, b) \in P_\epsilon \mid b = g(a)\right\} > dm + \ell.$$

(*Note:* Size of the input is around $\ell m \log n$. So, your time-complexity should be polynomial in that.)

□□□