CPS630: Homework 6

Your submission must be written up in LaTeX.

Problem 1: Two rooted trees T_1 and T_2 are said to be isomorphic if there is a one-to-one mapping f from the vertices of T_1 to the vertices of T_2 satisfying the following: The root of T_1 maps to the root of T_2. For each internal vertex $v \in T_1$ with children $v_1, v_2, \ldots, v_k$, the vertex $f(v)$ has children exactly $f(v_1), \ldots, f(v_k)$. Give a randomized fingerprinting algorithm to decide if two trees are isomorphic. For this purpose, associate a polynomial P_v with each vertex $v \in T$. For every leaf, $P = x_0$. For a node v at height h (where the height of a node is the distance to the furthest child leaf), and children $v_1, v_2, \ldots, v_k$, associate a variable x_h and define

$$P_v(\vec{x}) = (x_h - P_{v_1}(\vec{x}))(x_h - P_{v_2}(\vec{x})) \cdots (x_h - P_{v_k}(\vec{x})).$$

Note that the variable x_h associated with v only depends on the height of v, so that all vertices with the same height are associated with the same variable. Show by induction that if two trees are isomorphic, the polynomials defined above are identical, and hence derive the randomized algorithm.

Problem 2: Given two multi-sets S_1 and S_2 of integers, use polynomial identity testing to devise a linear time algorithm to check if these multi-sets are the same. Note that the standard sorting approach will take $O(n \log n)$ time, where $n = |S_1| = |S_2|$. Discuss the relative merits of the two approaches in terms of the sizes of the numbers being operated on.

Problem 3: How can we convert an algorithm that decides the *existence* of a perfect matching to one that actually finds the matching? What is the running time assuming that the algorithm for deciding the existence of a matching runs in time $T(n, m)$ on an n-vertex graph with m edges?

Problem 4: In this problem, we will devise a different fingerprinting algorithm for pattern matching. We will map any bit string s to a 2×2 matrix $M(s)$, as follows.

- For the empty string ϵ, $M(\epsilon) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.
- $M(0) = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$.
- $M(1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$.
- For non-empty strings x, y, we have $M(xy) = M(x)M(y)$.

  Show that this function has the following properties.

1. $M(x)$ is well-defined for all binary strings.
2. $M(x) = M(y)$ implies $x = y$.
3. For bit strings x of length n, the entries of $M(x)$ are bounded above by the n^{th} Fibonacci number.

By considering the matrix $M(x)$ modulo a suitable prime p, show how you would perform efficient randomized pattern matching. How would you implement this algorithm?

Problem 5: Let $Q(x_1, x_2, \ldots, x_n)$ be a multi-variate polynomial over a finite field $\mathbf{Z}_p$ with degree sequence $(d_1, d_2, \ldots, d_n)$. The degree sequence is defined as follows: Let $d_1 > 0$ be the maximum exponent of x_1 in Q and let $Q_1(x_2, \ldots, x_n)$ be the coefficient of $x_1^{d_1}$ in Q; let $d_2 > 0$ be the maximum exponent of x_2 in Q_1 and $Q_2(x_3, \ldots, x_n)$ be the coefficient of $x_2^{d_2}$ in Q_1; and so on.

Let $S_1, S_2, \ldots, S_n \subset \mathbf{Z}_p$ be arbitrary subsets. For $r_i \in S_i$ chosen uniformly and independently at random, show that

$$\Pr[Q(r_1, \ldots, r_n) = 0 \mid Q \neq 0] \leq \sum_{i=1}^{n} \frac{d_i}{|S_i|}.$$