CS 574, Spring 2021
Timothy Chan

Homework 4 (due April 19 Monday 5pm (CT))

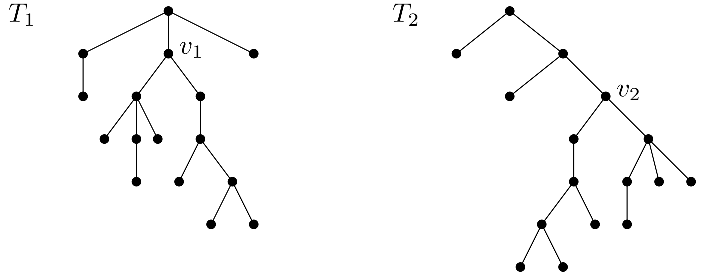
Instructions: You may work in groups of at most 3; submit one set of solutions per group. Always acknowledge any discussions you have with other people and any sources you have used (although most homework problems should be doable without using outside sources). In any case, *solutions must be written entirely in your own words.*

1. [*25 pts*] Given a string $s \in \Sigma^*$ of length n, and given an integer k, we want to find the longest substring t such that t occurs at least twice, and two occurrences are separated by at least k characters. (In other words, s contains tzt for some z of length at least k.)

   For example, for $s = 1100100100100110$ and $k = 5$, one answer is 1001 (by writing $s =$ 1**1001**00100**1001**10).

   Describe an efficient randomized (Monte Carlo) algorithm for this problem, by using the Karp–Rabin fingerprint technique. For full credit, the running time should be $O(n \log n)$ or better.

2. [*25 pts*] We are given two unordered rooted trees T_1 and T_2. Let $T(v)$ denote the subtree rooted at node v. We want to find two nodes $v_1 \in T_1$ and $v_2 \in T_2$ such that the subtree $T(v_1)$ is equivalent to the subtree $T(v_2)$, maximizing the subtree size. Here, in defining "equivalence" of trees, the children of each node is ignored (i.e., they may be re-ordered). For example, in the picture below, $T(v_1)$ and $T(v_2)$ are equivalent.

   Describe a randomized (Monte Carlo) algorithm that solves this problem in linear time.

   [*Hint*: for a node v with children $v_1, \ldots, v_d$, define a multivariate polynomial P_v recursively:

   $$P_v(x_1, \ldots, x_{h(v)}) \ := \ (x_{h(v)} - P_{v_1}(x_1, \ldots, x_{h(v_1)})) \cdots (x_{h(v)} - P_{v_d}(x_1, \ldots, x_{h(v_d)})),$$

   where $h(v)$ denotes the height of $T(v)$.]

3. [*23 pts*] We are given a bipartite graph G with n vertices, where each edge is colored red or blue. We are also given an integer k. Describe a randomized (Monte Carlo) $\widetilde{O}(n^\omega)$-time algorithm to decide whether G has a perfect matching with exactly k red edges. Here, $\omega < 2.38$ denotes the matrix multiplication exponent (and $\widetilde{O}$ hides polylogarithmic factors).

Your algorithm does *not* have to output such a matching. You may assume the following result: given an $n \times n$ matrix M where each entry is a polynomial in one variable x of degree at most d, with $O(d)$ coefficients from $\mathbb{Z}_p$, the determinant of M (which is a polynomial in x of degree at most dn, with $O(dn)$ coefficients from $\mathbb{Z}_p$) can be computed in $O(dn^\omega \log^{O(1)}(np))$ time for any given prime p.

4. [*27 pts*] In this problem, you will explore other simple families of hash functions and see how close they are to being universal. Let $U \geq m$.

   (a) [*9 pts*] Let p be a random prime in $[m/2, m]$. Let b be a random number in $\{0, \ldots, p-1\}$. Define $h_p : \{0, \ldots, U-1\} \to \{0, \ldots, m-1\}$ by

   $$h_p(x) = x \bmod p.$$

   Prove that for any fixed $x, y \in \{0, \ldots, U-1\}$ with $x \neq y$, we have $\Pr_p[h_p(x) = h_p(y)] \leq O((\log U)/m)$.

   (So, this hash function family may be a logarithmic-factor worse than being universal.)

   [*Note*: By the prime number theorem, there are $\Theta(m/\log m)$ primes in $[m/2, m]$.]

   (b) [*9 pts*] Suppose $U = 2^w$ and $m = 2^\ell$.

   Let M be a random 0-1 matrix of dimension $\ell \times w$. Define $h_M : \{0,1\}^w \to \{0,1\}^\ell$ by

   $$h_M(x) = (Mx) \bmod 2.$$

   All arithmetic operations are done modulo 2; for example, addition is equivalent to exclusive-or.

   Prove that for any fixed $x, y \in \{0,1\}^w$ with $x \neq y$, we have $\Pr_M[h_M(x) = h_M(y)] = 1/m$.

   (So, this hash function family is universal, and its computation avoids integer division and requires only bitwise exclusive-or!)

   (c) [*9 pts*] Continuing part (b), let b be a random 0-1 vector of dimension ℓ. Define $h_{M,b} : \{0,1\}^w \to \{0,1\}^\ell$ by

   $$h_{M,b}(x) = (Mx + b) \bmod 2.$$

   Prove that for any fixed $x, y \in \{0,1\}^w$ with $x \neq y$ and for any fixed $s, t \in \{0,1\}^\ell$, we have $\Pr_{M,b}[(h_{M,b}(x) = s) \wedge (h_{M,b}(y) = t)] = 1/m^2$.

   (So, this hash function family is strongly 2-universal.)