Practice Problems 1

Problem 1-1. Sub-linear time algorithms

(a) Given a string of 0's and 1's of size n, design an algorithm as efficient as possible that estimates the fraction of 1's in the string up to an additive ϵn.

(b) Extend the minimum spanning tree sub-linear time algorithm to handle real weights between 1 and w.

Problem 1-2. Pseudorandom Path of Length k

In recitation we considered the problem of finding a simple path with k edges in a given graph. We saw a randomized algorithm for this problem:

- Repeat the following $100k!$ times:
  1. Pick a random order on the vertices. Remove all edges that go backwards.
  2. Use dynamic programming to find the longest simple path in the graph. If it has at least k edges, return a sub-path with k edges.

Derandomize this algorithm using a "k-wise independent hash family." This is a hash family $H = \{h : V \to \{1, .., k\}\}$ such that for every different $v_1, \ldots, v_k$ and every vector b in $\{1, .., k\}^k$, the probability that a random $h \in H$ satisfies $h(v_1) = b_1 \wedge \ldots \wedge h(v_k) = b_k$ is k^{-k}.

Problem 1-3. Polynomial testing

It can be proved that an m-variate polynomial $P(x_1, \ldots, x_m)$ of total degree at most D that is not identically 0 has at most Dp^{m1} zeros modulo p in any set of inputs of the form Z_p^m for a prime p (recall that $Z_p = 0, ..., p - 1$).

Show how to find, given the coefficients of P, a point $(x_1, \ldots, x_m)$ that is a non-zero of P:

(a) Devise a randomized algorithm.

(b) Use the method of conditional expectations to find a deterministic algorithm.

Problem 1-4. Computational Geometry

(a) (CLRS Exercise 33.2-6) A *disk* consists of a circle plus its interior and is represented by its center point and radius. Two disks intersect if they have any point in common. Give an $O(n \lg n)$ time algorithm to determine whether any two disks in a set of n intersect.

(b) (CLRS Exercise 33.4-4) Given two points p_1 and p_2 in the plane, the L_∞-distance between them is given by $\max(|x_1 - x_2|, |y_1 - y_2|)$. Modify the closest-pair algorithm to use the L_∞-distance.