Sublinear Time Algorithms	December 1, 2020

Homework 5

Lecturer: Ronitt Rubinfeld	*Due Date: Not due.*

These problems are all extra credit.

1. Give a lower bound on computing a multiplicative estimate on the MST of a graph G in adjacency list representation: Give two distributions over graphs of degree at most d and weights in the range $\{1, \ldots, w\}$ (for $w = o(n)$) such that

   (a) graphs in one distribution have an MST weight that is at least twice the MST weight of the graphs in the in other distribution

   (b) in order to distinguish the two distributions with constant probability of success, one must make at least $\Omega(w)$ queries

   If you can get the lower bound to have some nontrivial dependence on d and ϵ, even better!

   (Note: It is possible to write this lower bound without explicitly using Yao's method.) You may assume that G is in adjacency list representation.

2. Suppose an algorithm has the following behavior when given error parameter ϵ and access to samples of a distribution p over a domain $D = \{1, \ldots, n\}$:

   - if p is monotone ($p(i) \geq p(i+1)$ for all i), then $\mathcal{A}$ outputs "pass" with probability at least 2/3.
   - if for all monotone distributions q over D, $|p - q|_1 > \epsilon$, then $\mathcal{A}$ outputs "fail" with probability at least 2/3

   Show that this algorithm must make $\Omega(\sqrt{n})$ queries.

   Hint: Reduce from the problem of testing uniformity.

3. In the following questions, assume that all input graphs start out with unique IDs.

   (a) Given a graph of max degree at most Δ, show that the edges can be decomposed into at most Δ oriented forests (where each node has outdegree at most 1, the roots have outdegree 0, and edges point along the path to a root). Show that given a node, the edge in oriented forest i and the direction of the edge, can be computed in $O(\Delta)$ sequential time.

   (b) Give a distributed algorithm for 6-coloring trees. Assume that the tree can be viewed as a rooted tree in which children know who their parent is. For full credit, your algorithm should run in $k = O(\log^* n)$ rounds. Note that this gives an LCA for 6-coloring trees which runs in $2^{O(\log^* n)} = O(\log^* n)$ probes. *Hint: Consider algorithms in which a node u looks at its parent v and recolors itself based on the location of the first bit which differs between u and v.*

   (c) Given graph G along with a c-coloring of the nodes (assume you can query the coloring of any node in 1 step). Show how to find an MIS in c distributed rounds.

   (d) Combine the above to give an LCA for 6^{Δ} coloring a degree at most Δ graph G.