Sublinear Time Algorithms	September 28, 2022

Homework 2

Lecturer: Ronitt Rubinfeld	*Due Date: October 12, 2022*

Turn in your solution to each *problem on a separate sheet of paper, with your name on each one.*

1. **Testing the monotonicity of a list – the case of bits:**

   Given a Boolean function $f : [n] \rightarrow \{0, 1\}$ and parameter $\epsilon \in (0, 1)$, present an algorithm that makes $1/\text{poly}(\epsilon)$ queries to f, and has the following behavior:

   - If f is monotone, then the algorithm always outputs "pass."
   - If f is ϵ-far from monotone, then the algorithm outputs "fail" with probability at least 3/4.

   Here by "ϵ-far from monotone" we mean that the value of f need only be changed on at most ϵn inputs in order to make it monotone.

2. **Removing adaptivity:**

   Assume that your computational model is such that a query returns a single bit. In such a model, show that any algorithm making q queries can be made into a *nonadaptive* (i.e., where the queries do not depend on the results of any previous queries) tester that uses only 2^q queries.

3. **Removing adaptivity for property testing dense graphs:**

   We define a *graph property* to be a property that is preserved under graph isomorphism (i.e., if Π is a graph property, then for any isomorphic graphs G and G', G has property Π if and only if G' has property Π).

   Show that any adaptive algorithm for testing a given graph property which makes q queries can be made into a nonadaptive algorithm for testing the same graph property using only $O(q^2)$ queries.

   Hint 1 : Prove that a q-query tester can be turned into a $O(q^2)$-query tester which tests all edges of some (possibly adaptively chosen) induced subgraph of the input graph G.

   Hint 2: Instead of running a tester on the original graph G, what would happen if you ran the tester on some isomorphic copy of G?

   Hint 3: Your nonadaptive algorithm is allowed to be randomized.

4. **Property testing of the clusterability of a set of points:**

   Let X be a set of points in an arbitrary metric space. Assume that one can compute the distance between any pair of points in one step. Say that X is (k, b)-diameter clusterable if X can be partitioned into k subsets, which we call "clusters," such that the maximum distance between any pair of points in a cluster is b. Say that X is ϵ-far from (k, b)-diameter clusterable if at least $\epsilon|X|$ points must be deleted from X in order to make it (k, b)-diameter clusterable.

Show how to distinguish the case where X is (k, b)-diameter clusterable from the case where X is ϵ-far from $(k, 2b)$-diameter clusterable. Your algorithm should use $\text{poly}(k, 1/\epsilon)$ queries. Note that it is possible to get an algorithm which uses $O((k^2 \log k)/\epsilon)$ queries.