| | |
|---|---|
| **Sublinear Time Algorithms** | March 4, 2019 |

# Homework 2

*Lecturer: Ronitt Rubinfeld* — *Due Date: March 18, 2019*

*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 function $f : [n] \to \{0,1\}$. Given $0 < \epsilon < 1$, show an algorithm that runs in $O(1/\text{poly}(\epsilon))$ queries to $f$, with the following behavior:
   - If $f$ is monotone, then the algorithm always outputs "pass".
   - If $f$ is $\epsilon$-far from monotone, then the algorithm outputs "fail" with probability at least $3/4$.

2. **How much can adaptivity help?**
   - 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.
   - *Canonical forms for graph property testers for the adjacency matrix model.* Define a graph property to be a property that is preserved under graph isomorphism – i.e., if $G$ has the property and $G'$ is isomorphic to $G$, then $G'$ must also have the property. Show that any adaptive algorithm for property testing which makes $q$ queries, can be made nonadaptive algorithm using only $O(q^2)$ queries.

3. **Property testing of the clusterability of a set of points.** Given a set $X$ of points in any 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 (clusters) such that the maximum distance between any pair of points in a cluster is $b$. Say that $X$ is $\epsilon$-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 when $X$ is $(k, b)$-diameter clusterable from the case when $X$ is $\epsilon$-far from $(k, 2b)$-diameter clusterable. Your algorithm should use polynomial in $k, 1/\epsilon$ queries. It is possible to get an algorithm which uses $O((k^2 \log k)/\epsilon)$ queries.