Problem Set (updated)

Problems

1. A *tournament* is a directed graph such that for any two distinct vertices u and v, there is either an edge from u to v, or an edge from v to u, but not both. Suppose an n-vertex tournament G is represented by a $n \times n$ matrix in which an entry (u, v) is 1 if G contains the edge (u, v) and -1 otherwise (that is, if G contains the edge (v, u)). A *sink* is a node u such that u's row contains only 1s.

   Give a deterministic algorithm to find a sink in a tournament, if it exists, in $O(n)$ queries.

2. In class we saw a tester for sortedness that used 2-spanners. In this problem we will design an alternative tester for sortedness that uses binary search.

   We are given a list of numbers $y_1, y_2, \ldots y_n$. Consider the following test. Pick a random i. Look at the value of y_i. Perform binary search for y_i. If the search finds any inconsistencies, then output FAIL. If the search does not end up at location i, then also output FAIL. Repeat this test $O(1/\epsilon)$ times. If none of them outputs FAIL, then output PASS.

   (a) Why does this test work?

   (b) What is the query complexity and running time of this test?

   (c) Show how to make the above test non-adaptive

   (d) As specified, the above algorithm only works for testing strict sortedness. Why is this so? Modify the algorithm to also work for testing non-strict sortedness.

3. We say that a function $f : D \to R$ is Lipschitz if $|f(x) - f(y)| \leq |x - y|$ for all $x, y \in D$. Recall that in class we showed how to test if a list is sorted using the spanner-based test. Come up with a similar proof to test the Lipschitz property for functions $f : \{1, 2, \ldots, n\} \to \mathbb{R}$. As a first step, show that the Lipschitz property allows extension for the range $\mathbb{R}$.

   Does the spanner-based test for the Lipschitz property apply if the range is $\mathbb{R}^2$ with Euclidean distances? How about the range $\mathbb{Z}^2$ with Euclidean distances?

4. Let $f : \{0,1\}^n \to \{0,1\}$ be a function. For an edge (x, y) in the boolean hypercube, with $x < y$, we say the edge is violated if $f(x) > f(y)$. Let $V(f)$ denote the number of edges violated by f. In class we showed the repair lemma. Namely, f can be made monotone by changing $\leq 2V(f)$ values. Show how to improve the bound obtained in the repair lemma by a factor of 2.

5. Let $f : \{0,1\}^n \to \{1, 2, \ldots, k\}$ be a function. Give a tester for testing the Lipschitz property for f that runs in time $O(d \cdot \min(d, k/\epsilon))$

   Hint: the tester should follow the outline for the tester for monotonicity on the hypercube. Transform f into a Lipschitz function by repairing the violated edges one dimension at a time. Repair edges by replacing the end points with the average of the original values at the end points (care needs to be taken to maintain integer values). Define a suitable notion of "magnitude of violation" of an edge, and show that when a particular dimension is repaired, then the magnitude of violation in the other dimensions does not increase.

6. This is an exercise in using Chernoff-Hoeffding Bounds.

   (a) [**Amplification of the success probability**] Given an algorithm A that produces a good approximation (additive or multiplicative) with probability at least $\frac{2}{3}$ and runs in time $T(n)$ on inputs of size n, convert it to an algorithm that produces a good approximation (in the same sense as A) with error probability $\leq \delta$ and runs in time $O(T(n) \log \frac{1}{\delta})$. (Hint: run the algorithm $\Theta(\log \frac{1}{\delta})$ times and output the median answer.) Note that for algorithms with 0/1 output (e.g., property testers), taking the median corresponds to taking the majority.

   (b) [**Witness observation extension**] Suppose that an α fraction of vertices in a graph are "witnesses". Prove that a sample of $s = \max\left\{\frac{2m}{\alpha}, \frac{8\ln(1/\delta)}{\alpha}\right\}$ vertices (selected uniformly and independently) will have at least m "witnesses" with probability at least $1 - \delta$.

7. **A property tester for connectedness** We are given a graph $G = (V, E)$, where $|V| = n$ and $|E| = m$. G has no self-loops. The degree of G is bounded by d, and G is given in adjacency list format. Design an algorithm to test whether G is connected. More formally, let $P_{n,d} = \{G' :$ G' is a connected n vertex graph with max degree at most d$\}$. We say that G is ϵ-far from $P_{n,d}$ if we must add/delete at least ϵdn edges to make it a member of $P_{n,d}$. Design a tester so that if G is connected, then probability of outputting PASS is $> 2/3$, and if G is ϵ-far from conenected, then probability of outputting FAIL is $> 2/3$.

   Hint: If G is far from connected, then it will have a lot of connected components, which means that it will have a lot of small connected components. Hence with reasonable probability, a random vertex will land in a small connected component.

8. Given a list $x_1, x_2, \ldots, x_n$, where each $x_i \in \{0, 1\}$, show how to test if the list is sorted using $O(\frac{1}{\epsilon})$ queries.

9. An *image* is an $n \times n$ matrix of 0/1 pixel values where 0 represents white and 1 represents black. To keep the correspondence with the plane, index the matrix by $\{0, 1, \ldots, n-1\}^2$, with the lower left corner being $(0, 0)$ and the upper left corner being $(0, n-1)$.

   (a) The *image graph* $G_M = (V, E)$ of an image matrix M has vertex set $V = \{(i, j) | M_{ij} = 1\}$ and edge set $E = \{((i_1, j), (i_2, j)) \mid |i_1 - i_2| = 1\} \cup \{((i, j_1), (i, j_2)) \mid |j_1 - j_2| = 1\}$. In other words, the image graph consists of black pixels connected by the grid lines. The image is *connected* if its image graph is connected.

   Adopt the tester for connectedness of graphs to test for connectedness of images. (Be careful: the tester is almost the same, but its analysis is different because the distance between two images is measured differently than the distance between two graphs in the bounded degree model.)

10. **Yao's Principle:**

    To show that every probabilistic algorithm needs $q = q(n, \epsilon)$ queries to test property $\mathcal{P}$ on inputs of length n with probability of error $\leq \frac{1}{3}$ (over the algorithm's coin tosses), it is enough to show that there is a distribution $\mathcal{D}$ on the inputs of length n such that every deterministic algorithm needs q queries to test $\mathcal{P}$ over inputs from $\mathcal{D}$ with probability of error $\leq \frac{1}{3}$ (over the choice of the input).

    Prove this statement for 2-sided error adaptive algorithms. Notice that the same reasoning applies to 1-sided error (algorithms that accept inputs which have the property with probability 1) and/or non-adaptive algorithms (algorithms that decide what points to query right at the beginning, without waiting to see the values at the queried points).

11. Use Yao's Principle to give a lower bound of $\Omega(\log n)$ for testing sortedness of a list that applies to non-adaptive one-sided error testers. (The spanner-based tester discussed in class is an example of such a tester for sortedness).

12. Prove a lower bound for testing monotonicity of a function $f : \{0, 1\}^n \to \mathbb{R}$ by giving a reduction from a communication complexity problem.

Instructions: Some of these problems are quite tricky and their solutions have resulted in actual research papers. Don't be discouraged if they look difficult. Think about the problems at home. Many of them will be discussed in the review sessions. If you want hints for any of the problems, then just ask. If you would like to write up solutions to any of the problems, then they will be graded and handed back to you. Collaboration is allowed and encouraged!