6.5240 Midterm Solutions Fall 2024

1. **(8 points.) Testing the monotonicity of a list – the case of bits:** Given a Boolean function $f : [n] \to \{0, 1\}$ and parameter $\varepsilon \in (0, 1)$, present an algorithm that makes $1/\text{poly}(\varepsilon)$ 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.

   For a bonus point, get a query complexity of $O(\frac{1}{\varepsilon} \cdot \log(1/\varepsilon))$.

2. **(8 points) Lower bounds for estimating MST weight:** Give a lower bound on computing a multiplicative estimate on the MST of a graph G in adjacency list representation (i.e. neighbor queries, degree queries, and edge weight queries in one step): 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

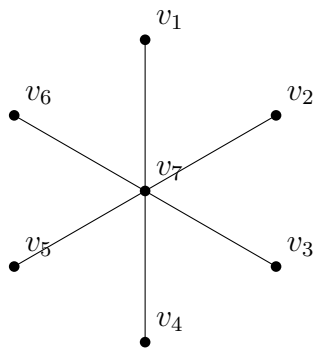
   b) in order to distinguish the two distributions with constant probability of success, one must make at least $\Omega(w)$ queries

   There is a simple construction that works. For a bonus point, show you can get the lower bound to have some nontrivial dependence on d.

3. **(8 points.) Counting wedges.** In this problem we will create an algorithm for counting the number of *unlabeled wedges* (2-stars, or undirected paths of length 2) in a simple graph $G = (V, E)$ with n vertices and m edges. The access model is that the algorithm can ask for any of the following in one step: a uniform random edge, a uniform random neighbor of a given vertex, and the degree of a given vertex.

   Please see the following figure for an example of how wedges are counted.

Figure 1: In this example, there are $\binom{6}{2} = 15$ wedges, each with v_7 as the center. For instance, (v_6, v_7, v_3) is a wedge (and (v_3, v_7, v_6) is the same wedge), but (v_1, v_7, v_1) is not because it contains repeated vertices.



a) **(1 point.)** Give an expression for the number of *wedges* in G as a function of the degrees.

b) **(1 points.)** Let w be the number of wedges in G. Assume you know some bound $\overline{w}$ that is at least $w/2$ and $\overline{w} \geq 1$. Show that no vertex has degree greater than $d_{ub} := \min(n, 3\sqrt{\overline{w}})$

c) **(3 points.)** Give an algorithm that returns each wedge with probability $1/(m \cdot d_{ub})$ (or else returns $\perp$) and makes $O(1)$ queries.

   Hint: Consider first sampling an edge uniformly at random.

d) **(3 points.)** Assuming there is an algorithm satisfying (c), give an algorithm that returns a $(1 \pm \varepsilon)$-approximation of w and makes

$$O\left(\varepsilon^{-2} \cdot \frac{m d_{ub}}{\overline{w}}\right)$$

   queries, assuming knowledge of $\overline{w}$. Note that because of the bounds on d_{ub}, this complexity is equivalently

$$O\left(\varepsilon^{-2} \min\left(\frac{mn}{\overline{w}}, \frac{m}{\sqrt{\overline{w}}}\right)\right).$$

   The failure probability can be a small constant, such as $1/4$.

Blank page