Homework 4 – Due Thursday, April 3 *before* 11am on Gradescope

Instructions

- Solutions written in LaTeX are strongly preferred, but you can upload any pdf files, including scanned hand-written solutions. Template latex files are on the course webpage.
- Collaboration is allowed and encouraged. However, each of you should think about a problem before discussing it with others and write up your solution independently. You may consult books and online sources to get information about well-known theorems, such as the Chernoff bound. But you are not allowed to look up solutions directly in papers or any other sources. You *must* list all collaborators and sources! (See full details in the General Information Handout.)
- Correctness, clarity, and succinctness of the solution will determine your score.

Problems

1. This is a collection of questions with short answers on testers in the adjacency matrix model.
   (a) Based on our analysis in class, give a new ϵ-test for bipartiteness with query and time complexity $O(\frac{\log^2 1/\epsilon}{\epsilon^3})$ instead of $O(\frac{\log^2 1/\epsilon}{\epsilon^4})$. Provide a one-line explanation.
   (b) In class we saw an ϵ-additive approximation algorithm for the edge density of the max cut with running time exponential in $\frac{1}{\epsilon}$. Explain why it is unlikely that there is such an algorithm with $poly(\frac{1}{\epsilon})$ running time that works for all ϵ. (Hint: it would imply NP $\subseteq$ BPP.)
   (c) Improve the running time of the probabilistic algorithm (from class) that finds a cut with density at least $\mu(G) - \epsilon$ from $2^{poly\frac{1}{\epsilon}} n^2$ to $2^{poly\frac{1}{\epsilon}} + O(n \cdot poly\frac{1}{\epsilon})$. Explain the new algorithm and give a one-line justification. Hint: we proved all the necessary lemmas in class.

2. (**An Application of the Regularity Lemma**) In this problem, you are asked to show that the number of triangle-free labeled graphs on n nodes is $2^{\left(\frac{1}{4}+o(1)\right)n^2}$, as indicated in the following parts.
   (a) Show that there are at least $2^{\frac{1}{4}n^2}$ triangle-free labeled graphs on n nodes, where $n > 1$. (Make sure you consider both odd and even n.)
   (b) Prove (e.g., by induction) that every triangle-free graph contains at most $\frac{n^2}{4}$ edges.
   (c) Using the Regularity Lemma and 2b, show that there are at most $2^{\left(\frac{1}{4}+o(1)\right)n^2}$ triangle-free labeled graphs on n nodes.
   Hint: Consider a sufficiently large triangle-free graph G and a tiny $\epsilon > 0$. Equipartition the vertices of G into $\frac{1}{\epsilon}$ sets, apply the Regularity Lemma, and carefully count the number of possibilities.
   You may use the following facts without a proof:

$$\binom{n}{\alpha n} \leq 2^{n(H(\alpha)+o(1))} \text{ where } H(\alpha) = -\alpha \log_2 \alpha - (1-\alpha)\log_2(1-\alpha). \quad (1)$$

$$\text{The entropy function } H(\alpha) \text{ tends to 0 as } \alpha \text{ tends to 0.} \quad (2)$$

3. (**Linearity Testing**) Consider Algorithm 1, which is a generalization of the BLR linearity test.

Algorithm 1: $\textsc{LinearityTest}_k$

Input: Even integer parameter $k \geq 2$ and query access to a function $f : \{-1,1\}^n \to \{-1,1\}$

Query k points $x^{(1)}, \ldots, x^{(k)} \in \{-1,1\}^n$ chosen uniformly at random (with replacement).

Query point $y = \sum_{i \in [k]} x^{(k)}$.

Reject if $f(y) \neq \prod_{i \in [k]} f(x_i)$; otherwise, **accept**.

(a) Let k be even and $f : \{-1,1\}^n \to \{-1,1\}$ be a function that is ϵ-far from linear. Prove that the probability that $\textsc{LinearityTest}_k(f)$ rejects is

$$\frac{1}{2} - \frac{1}{2} \sum_{S \subseteq [n]} \widehat{f}(S)^{k+1}.$$

(b) Argue that

$$\sum_{S \subseteq [n]} \widehat{f}(S)^{k+1} \leq (1 - 2\epsilon)^{k-1}.$$

(c) How many iterations of Algorithm 1 are needed to get a linearity tester that has probability of error at most $\frac{1}{3}$? How does your linearity tester compare to the BLR tester in terms of query complexity?

In your explanation for this part, you can use the following fact without a proof:

$$\frac{1 - (1 - 2\epsilon)^{k-1}}{2} \geq \min\left\{\frac{1}{4}, \frac{k\epsilon}{2}\right\}.$$

(d) Would your algorithm from the previous part work if k were odd?