Homework 3 – Due Thursday, October 8 *before* 10am 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. And 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. (**Lipschitz functions with a small range**) Recall that a function $f : \{0,1\}^d \to R$ is called *Lipschitz* if for d-bit strings x and y that differ in one bit, $|f(x) - f(y)| \leq 1$.
   (a) Give an ϵ-tester for the Lipschitz property of functions of the form $f : \{0,1\}^d \to \{0,1,2\}$. Your tester should run in time $O\left(\frac{d}{\epsilon}\right)$.
   (b) Prove by giving a reduction from a communication complexity problem that every $\frac{1}{4}$-tester for the Lipschitz property of functions of the form $f : \{0,1\}^d \to \{0,1,2\}$ must make $\Omega(d)$ queries.
2. (**Yao's Principle**) Formulate the property testing version of the problem of checking whether all characters in the input string are distinct. To keep with our streaming notation, represent your input as an m-character sequence over alphabet $[n]$. Use Yao's Principle to show that every $\frac{1}{2}$-tester for this property must read $\Omega(\sqrt{m})$ input characters.
3. (**Set Intersection**) Recall the streaming algorithm for estimating the number of distinct elements from class. It can be viewed as first producing a sketch of the data (containing t smallest values of the hash seen in the stream) and then returning the answer. The goal of this problem is to design an algorithm for estimating the size of the intersection of two streams over the same domain $[n]$, using the distinct elements sketch as a primitive. For example, if the streams are $\langle 7,1,2,2\rangle$ and $\langle 2,7,2,7,5\rangle$, then the intersection is $\{1,2,7\} \cap \{2,5,7\} = \{2,7\}$, and its size is 2.

   In particular, you get to specify the parameters of the distinct elements estimation algorithm you want to run on the streams (e.g., how many instantiations to run, which hash functions to use, etc.) After your algorithm runs on two different streams, you will get the two resulting sketches. You need to explain how to postprocess them to estimate the size of the intersection of the two streams. Let $\epsilon, \delta \in (0, 1/2)$ be parameters given to your algorithm, and m be the total number of elements in the two streams. You estimate should have error $\pm\epsilon m$ with probability $1-\delta$.

   Specify your algorithm, prove its correctness, and state its space complexity.