Carnegie Mellon University Instructor: Venkatesan Guruswami
15-252: More GITCS Term: Spring 2021

Problem Set 11

Due date: Friday, April 30 (by midnight EDT)

Instructions

- You are allowed to collaborate with one other student taking the class, or do it solo.
- Collaboration is defined as discussion of the lecture material and solution approaches to the problems. Please note that *you are not allowed to share any written material and you must write up solutions on your own.* You must clearly acknowledge your collaborator in the write-up of your solutions.
- Solutions must be submitted on gradescope. Typesetting in LaTeX is recommended but not required. If submitting handwritten work, please make sure it is a legible and polished final draft.
- You should not search for solutions on the web. More generally, you should try and solve the problems without consulting any reference material other than what we cover in class and any provided notes.
- Please start working on the problem set early. Though it is short, the problem(s) might take some time to solve.

---

1. Suppose that a probability distribution Π on $\{0,1\}^n$ is (C,ϵ)-pseudorandom, i.e., for every Turing Machine M running in time $C(n)$,

$$\left| \ \mathbf{Pr}_{x\sim\Pi}\ [M(x)=1] - \mathbf{Pr}_{x\in U_n}\ [M(x)=1] \ \right| \leq \epsilon$$

where U_n is the uniform distribution on $\{0,1\}^n$.

Prove that there exists no Turing Machine that runs in time $C(n)$ and predicts n'th bit of a sample from distribution Π from its first $(n-1)$ bits with accuracy better than $\frac{1}{2}+\epsilon$; that is, there is no function $g:\{0,1\}^{n-1}\to\{0,1\}$ computable by a Turing machine running in time $C(n)$ such that

$$\mathbf{Pr}_{x\sim\Pi}\ \left[\ g(x_1,x_2,\ldots,x_{n-1})=x_n\ \right] > \frac{1}{2}+\epsilon$$

2. In this exercise, we will prove that the least significant bit $\mathrm{lsb}(x)$ is not a hardcore predicate for the exponentiation function $g^x \mod p$. Let $p>3$ be a prime number. Let $Z_p^*=\{1,2,\ldots,p-1\}$. It is well known that there is an integer $g\in Z_p^*$ such that $Z_p^*=\{g,g^2,\ldots,g^{p-1}\}$ (here, all the multiplications are modulo p.) Prove that there is an algorithm that runs in time $\mathrm{poly}(\log p)$ that given g^x for some $x\in Z_p^*$ outputs correctly whether x is even or odd.

<u>Hint</u>: An element $z\in Z_p^*$ is a square iff $z^{(p-1)/2}\equiv 1 \pmod p$.