

6.5620 (6.875), Fall 2022

Homework # 1

Due: 21 September 2022, 11:59:59pm ET

- **Typsetting:** You are encouraged to use L^AT_EX to typeset your solutions. You can use the following [template](#).
 - **Submissions:** Solutions should be submitted to Gradescope.
 - **Reference your sources:** If you use material outside the class, please reference your sources (including papers, websites, wikipedia).
 - **Acknowledge your collaborators:** Collaboration is permitted and encouraged in small groups of at most three. You must write up your solutions entirely on your own and acknowledge your collaborators.
-

Problems:

1. (5 points) **Working with negligible functions.** Recall that a non-negative function $\nu : \mathbb{N} \rightarrow \mathbb{R}^+$ is *negligible* if it decreases faster than the inverse of any polynomial (otherwise, we say that ν is *non-negligible*). More precisely, for all $c > 0$, there exists a constant $N > 0$ such that for all $n \geq N$, $\nu(n) < n^{-c}$.

State whether each of the following functions is negligible or non-negligible, and prove your assertion. For all of the problems below, we take the base of the logarithm to be 2.

- (a) $\nu(n) = 1/2^{100 \log n}$.
- (b) $\nu(n) = 1/2^{100 \log n \cdot \log \log \log n}$.
- (c) $\nu(n) = p(n) \cdot \mu(n)$, where $p(n) = O(n^k)$ for some constant k , and $\mu(n)$ is a negligible function. Either prove that ν is always negligible, or come up with a counter-example.
- (d) $\nu(n) = (\mu(n))^{\frac{1}{p(n)}}$, where $p(n)$ and $\mu(n)$ are as defined in (c). Either prove that ν is always negligible, or come up with a counter-example.
- (e) $\nu(n) = 1/2^{\log^* n}$, where $\log^* n$ is the number of times the logarithm function must be iteratively applied to n before the result is less than or equal to 1. More concretely,

$$\log^* n := \begin{cases} 0 & \text{if } n \leq 1 \\ 1 + \log^*(\log n) & \text{if } n > 1. \end{cases}$$

You may use the fact that $\log^* n \leq n$ without proof. (Hint: How does $\log^* n$ compare to $\log \log n$?)

2. (11 points) **Statistical and computational indistinguishability.** We think of distributions X, Y on a (finite) set Ω as functions $X, Y : \Omega \rightarrow [0, 1]$ such that for $\sum_{\omega \in \Omega} X(\omega) = \sum_{\omega \in \Omega} Y(\omega) = 1$. The statistical distance (also known as variational or L_1 distance) between X and Y is defined as

$$\Delta(X, Y) := \frac{1}{2} \sum_{\omega \in \Omega} |X(\omega) - Y(\omega)|.$$

- (a) (3 points) Show that the following is an equivalent definition:

$$\Delta(X, Y) := \sup_{A \subseteq \Omega} |X(A) - Y(A)|,$$

where $X(A)$ is shorthand for $\sum_{\omega \in A} X(\omega)$.

- (b) (3 points) Let D_0 and D_1 be two distributions over Ω . Suppose that we play the following game with an algorithm \mathcal{A} . First, we pick at random a bit $b \leftarrow \{0, 1\}$ and then we pick $x \leftarrow D_b$ and we give x to \mathcal{A} . Finally, \mathcal{A} returns a bit. It wins if the bit returned is equal to b . Show that the highest success probability in this game is exactly $\frac{1}{2} + \frac{1}{2}\Delta(D_0, D_1)$.
- (c) (1 point) Give the definition of computational indistinguishability using similar language as in the previous question. (This part should not take more than 3-5 sentences.)
- (d) (4 points) For a probability distribution D over Ω and positive integer m , let D^m denote the *product distribution* over Ω^m , obtained by drawing a tuple of m independent samples from D . Let $\mathcal{X} = \{X_n\}_n$ and $\mathcal{Y} = \{Y_n\}_n$ be ensembles of distributions that are efficiently sampleable (in PPT), and let $m(n) = \text{poly}(n)$ be some fixed polynomial. Prove that if \mathcal{X} and \mathcal{Y} are computationally indistinguishable, or, in symbols, $\mathcal{X} \stackrel{c}{\approx} \mathcal{Y}$, then $\{X_n^{m(n)}\} \stackrel{c}{\approx} \{Y_n^{m(n)}\}$. (Where do you use that X_n, Y_n are efficiently sampleable?) This shows that if two efficiently sampleable distributions are computationally indistinguishable given one sample, then they are also computationally indistinguishable given polynomially many samples. (Your distinguisher is allowed to be *non-uniform*¹, but you must justify where you use this fact.)
3. (8 points) **PRG or not?** Let $G : \{0, 1\}^{2n} \rightarrow \{0, 1\}^\ell$ be a pseudorandom generator, where $\ell \geq 2n + 1$. In each of the following, say whether G_c is necessarily a pseudorandom generator. If yes, give a proof. Otherwise, show a counterexample. Your counterexamples must rely only on the existence of pseudorandom generators.

¹A non-uniform polynomial-time distinguisher is allowed, in addition to its input samples, to take as ‘advice’ a polynomial-length string $a(n)$, where the advice string a can depend arbitrarily on n . This may be useful if the advice string a is information (which depends only on n) that the distinguisher cannot necessarily compute efficiently—or at all—but which can be expressed efficiently. For more details, see Definition 6.9 of [Arora and Barak](#).

- (a) (2 points) Consider $G_0 : \{0, 1\}^{2n} \rightarrow \{0, 1\}^\ell$, where $G_0(s) := G(\bar{s})$. Here, \bar{s} is the bit-wise complement of s .
 - (b) (3 points) Consider $G_1 : \{0, 1\}^n \rightarrow \{0, 1\}^\ell$, where $G_1(s) := G(0^n || s)$.
 - (c) (3 points) Consider $G_2 : \{0, 1\}^{2n} \rightarrow \{0, 1\}^{2\ell}$, where $G_2(s) := G(s) || G(s + 1 \bmod 2^{2n})$.
4. (3 points) **A PRG from a PRF.** Prove that, if F is a length preserving pseudorandom function, then $G(s) \stackrel{\text{def}}{=} F_s(\langle 1 \rangle) || F_s(\langle 2 \rangle) || \dots || F_s(\langle \ell \rangle)$, where $\langle i \rangle$ is the n -bit binary representation of i , is a pseudorandom generator with expands ℓ bits to $\ell \cdot n$ bits.
5. (8 points) **Locking schemes.**

Veronica claims to Lucy, her classmate in 6.5620, that she can read minds, but Lucy does not believe it one bit, and is willing to bet \$100 that it's all baloney. They decide to play a game where Veronica can prove to Lucy that she is a bonafide mind-reader. If Veronica is lying, she should not be able to win in the game; and if Veronica indeed possesses this supernatural power, Lucy should grant victory (and \$100) to Veronica at the end of the game (no matter how much she hates to lose). They decide to play the following game.

- (a) Veronica (we'll call her V) sends Lucy a string v (of a certain length that they decided on), chosen at random.
- (b) Lucy (we'll call her L) chooses a random bit $\sigma \in \{0, 1\}$, and Veronica has to guess what this bit is. To do this, Lucy sends

$$\ell \leftarrow L(\sigma, v; r)$$

to Veronica, where r is Lucy's private random coins that only she knows.

- (c) Now, Veronica reads Lucy's mind and guesses what σ is.
- (d) Finally, Lucy "unlocks" her bit σ by sending σ and r to Veronica. Veronica verifies that ℓ is indeed $L(\sigma, v; r)$.

Veronica wins if her guess of σ is correct.

- (a) (1 point) Veronica should not gain any information about Lucy's bit from viewing the lock (i.e. after step 2 of the game). In other words, a malicious (but computationally *bounded*) Veronica V^* should not be able to learn anything about the honest L 's choice bit σ , no matter what initial message v^* she sent.

Using the notion of indistinguishability, give a formal definition of this *concealing property* of L .

- (b) (1 point) Lucy should not be able to unlock her bit both ways, otherwise she can always get away with not paying Veronica \$100 even if Veronica is a mind-reader (do you see why?) To ensure this, the locking algorithm L has to be "unmodifiable". Give a formal definition of this property.

- (c) (*2 points*) Let G be any length-tripling function, i.e., one for which $|G(x)| = 3|x|$ for every $x \in \{0, 1\}^*$. Give an upper bound on the probability, over the choice of a random $3n$ -bit string v , that there exist two inputs $x_1, x_2 \in \{0, 1\}^n$ such that $G(x_1) \oplus G(x_2) = v$.
- (d) (*4 points*) Let G be a length-tripling PRG (which we have seen can be obtained from any PRG). Use G to construct a secure locking scheme (i.e. define the algorithms V and L), and prove that it is both concealing and unmodifiable according to your definitions.