Problem 6.1. (Min-entropy and Statistical Difference)

1. Prove that for every two random variables $X$ and $Y$,
   \[
   \Delta(X,Y) = \max_f |E[f(X)] - E[f(Y)]| = \frac{1}{2} \cdot |X - Y|_1,
   \]
   where the maximum is over all $[0,1]$-valued functions $f$. (Hint: first identify the functions $f$ that maximize $|E[f(X)] - E[f(Y)]|$.)

2. Suppose that $(W,X)$ are jointly distributed random variables where $(W,X)$ is a $k$-source and $|\text{Supp}(W)| \leq 2^k$. Show that for every $\varepsilon > 0$, with probability at least $1 - \varepsilon$ over $w \overset{\text{iid}}{\rightarrow} W$, we have $X|_{W=w}$ is a $(k - \ell - \log(1/\varepsilon))$-source.

3. Suppose that $X$ is an $(n - \Delta)$-source taking values in $\{0,1\}^n$, and we let $X_1$ consist of the first $n_1$ bits of $X$ and $X_2$ the remaining $n_2 = n - n_1$ bits. Show that for every $\varepsilon > 0$, $(X_1,X_2)$ is $\varepsilon$-close to some $(n_1 - \Delta, n_2 - \Delta - \log(1/\varepsilon))$ block source.

Problem 6.5. (Extractors vs. Samplers) Use the connection between extractors and averaging samplers to do the following:

1. Prove that for all constants $\varepsilon, \alpha > 0$, there is a constant $\beta < 1$ such that for all $n$, there is an explicit $(\beta n, \varepsilon)$ extractor $\text{Ext} : \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^m$ with $d \leq \log n$ and $m \geq (1 - \alpha)n$.

2. Prove that for every $m \in \mathbb{N}$, $\varepsilon, \delta > 0$, there exists a (nonconstructive) $(\delta, \varepsilon)$ averaging sampler $\text{Smp} : \{0,1\}^n \rightarrow \{(0,1)^m\}^t$ using $n = m + 2\log(1/\varepsilon) + \log(1/\delta) + O(1)$ random bits and $t = O(1/(\varepsilon^2 \delta))$ samples.
3. Suppose we are given a constant-error BPP algorithm that uses \( r = r(n) \) random bits on inputs of length \( n \). Show how, using the explicit extractor of Theorem 6.36, we can reduce its error probability to \( 2^{-\ell} \) using \( O(r) + \ell \) random bits, for any polynomial \( \ell = \ell(n) \). (Note that this improves the \( r + O(\ell) \) given by expander walks for \( \ell \gg r \).) Conclude that every problem in BPP has a randomized polynomial-time algorithm that only errs for \( 2^{\varepsilon_0 \ell} \) choices of its \( q = q(n) \) random bits.

**Problem 6.7. (The Building-Block Extractor)** Prove Lemma 6.37: Show that for every constant \( t > 0 \) and all positive integers \( n \geq k \) and all \( \varepsilon > 0 \), there is an explicit \( (k, \varepsilon) \)-extractor \( \text{Ext}: \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^m \) with \( m \geq k/2 \) and \( d = k/t + O(\log(n/\varepsilon)) \). (Hint: convert the source into a block source with blocks of length \( k/O(t) + O(\log(n/\varepsilon)) \).)

**Problem 6.8. (Encryption and Deterministic Extraction)** A (one-time) encryption scheme with key length \( n \) and message length \( m \) consists of an encryption function \( \text{Enc}: \{0,1\}^n \times \{0,1\}^m \rightarrow \{0,1\}^\ell \) and a decryption function \( \text{Dec}: \{0,1\}^n \times \{0,1\}^\ell \rightarrow \{0,1\}^m \) such that \( \text{Dec}(k, \text{Enc}(k, u)) = u \) for every \( k \in \{0,1\}^n \) and \( u \in \{0,1\}^m \). Let \( K \) be a random variable taking values in \( \{0,1\}^n \). We say that \( (\text{Enc}, \text{Dec}) \) is (statistically) \( \varepsilon \)-secure with respect to \( K \) if for every two messages \( u, v \in \{0,1\}^m \), we have \( \Delta(\text{Enc}(K, u), \text{Enc}(K, v)) \leq \varepsilon \). For example, the one-time pad, where \( n = m = \ell \) and \( \text{Enc}(k, u) = k \oplus u = \text{Dec}(k, u) \) is 0-secure (aka perfectly secure) with respect to the uniform distribution \( K = U_m \). For a class \( C \) of sources on \( \{0,1\}^n \), we say that the encryption scheme \( (\text{Enc}, \text{Dec}) \) is \( \varepsilon \)-secure with respect to \( C \) if \( \text{Enc} \) is \( \varepsilon \)-secure with respect to every \( K \in C \).

1. Show that if there exists a deterministic \( \varepsilon \)-extractor \( \text{Ext}: \{0,1\}^n \rightarrow \{0,1\}^m \) for \( C \), then there exists an \( 2\varepsilon \)-secure encryption scheme with respect to \( C \).

2. Conversely, use the following steps to show that if there exists an \( \varepsilon \)-secure encryption scheme \( (\text{Enc}, \text{Dec}) \) with respect to \( C \), where \( \text{Enc}: \{0,1\}^n \times \{0,1\}^m \rightarrow \{0,1\}^\ell \), then there exists a deterministic \( 2\varepsilon \)-extractor \( \text{Ext}: \{0,1\}^n \rightarrow \{0,1\}^{m-2\log(1/\varepsilon) - O(1)} \) for \( C \), provided \( m \geq \log n + 2\log(1/\varepsilon) + O(1) \).

   (a) For each fixed key \( k \in \{0,1\}^n \), define a source \( X_k \) on \( \{0,1\}^\ell \) by \( X_k = \text{Enc}(k, U_m) \), and let \( C' \) be the class of all these sources (i.e., \( C' = \{X_k : k \in \{0,1\}^n\} \)). Show that there exists a deterministic \( \varepsilon \)-extractor \( \text{Ext}' : \{0,1\}^\ell \rightarrow \{0,1\}^{m-2\log(1/\varepsilon) - O(1)} \) for \( C' \), provided \( m \geq \log n + 2\log(1/\varepsilon) + O(1) \).

   (b) Show that if \( \text{Ext}' \) is a deterministic \( \varepsilon \)-extractor for \( C' \) and \( \text{Enc} \) is \( \varepsilon \)-secure with respect to \( C \), then \( \text{Ext}(k) = \text{Ext}'(\text{Enc}(k, 0^m)) \) is a deterministic \( 2\varepsilon \)-extractor for \( C \).

Thus, a class of sources can be used for secure encryption iff it is deterministically extractable.

**Problem 6.9. (Extracting from Symbol-Fixing Sources*)** A generalization of a bit-fixing source is a symbol-fixing source \( X \) taking values in \( \Sigma^n \) for some alphabet \( \Sigma \), where subset of the coordinates of \( X \) are fixed and the rest are uniformly distributed and independent elements of \( \Sigma \). For \( \Sigma = \{0,1\} \) and \( k \in [0, n] \), give an explicit \( \varepsilon \)-extractor \( \text{Ext}: \Sigma^n \rightarrow \{0,1\}^m \) for the class of of symbol-fixing sources on \( \Sigma^n \) with min-entropy at least \( k \), with \( m = \Omega(k) \) and \( \varepsilon = 2^{-O(k)} \). (Hint: use a random walk on a consistently labelled 3-regular expander graph.)