Problem 1 You are given a program $A$ which computes an unknown polynomial $P(x_1, \ldots, x_m)$ of degree $d$ over the field $\mathbb{Z}_p$ but makes an error on an $\varepsilon$ fraction of the inputs. Design a probabilistic program $B$ (which can use $A$ as a subroutine) that for EVERY input $x_1, \ldots, x_m$ computes $P(x_1, \ldots, x_m)$ with probability 99%. The running time of your program should be polynomial in $n$, where $n$ is the length of $p$ in binary (and $d$ and $m$ are bounded by polynomials of $n$; and $p$ is much larger than $d$ and $m$).

**Hint:** Consider a random line in $(\mathbb{Z}_p)^d$ and reduce the problem to the one dimensional case.

Problem 2 Let $g$ be a generator of the group $\mathbb{Z}_p^*$ ($p$ is a prime number) and let $f(x)$ be the discrete logarithm of $x$ (i.e. $g^{f(x)} = x \mod p$). Assume that

$$\Pr_{x \in \mathbb{Z}_p^*} (A(x) = f(x)) \geq \varepsilon,$$

for some algorithm $A$. Prove that for every $\delta$ there exists a probabilistic algorithm $B$ (which can use $A$ as a black box) such that for every $x$:

$$\Pr (B(x) = f(x)) \geq 1 - \delta.$$

Estimate the runtime of $B$ in terms of $\varepsilon$ and $\delta$.

**Hint:** Use the fact that exponentiation is easy to verify correctness of $A$.

Problem 3 Let $f(x)$ be as in Problem 2. Prove that computing the least significant bit of $f(x)$ is easy.

**Hint:** It depends on whether $x^{(p-1)/2}$ is 1 or -1.

Problem 4 Consider the following Hadamard matrix $H = (h_{st})$ indexed by binary strings (vectors) $s$ and $t$ of length $\ell$:

$$h_{st} = \langle s, t \rangle \mod 2 \equiv \sum_{i=1}^{\ell} s_i t_i \mod 2.$$

Note that the size of the matrix is $2^\ell \times 2^\ell$. In class we showed that this matrix can be used to derandomize a simple MAX CUT algorithm. Consider a program that picks a random string $s$ in $\{0,1\}^\ell$ and a random bit $b$ in $\{0,1\}$. Then it outputs the bits of the matrix in the row $s$ XOR (exclusive or) $b$. Prove that this program is not a pseudorandom generator.