Problem 5-1 (BPP and NP) 15 points
Show that if \( \text{NP} \subseteq \text{BPP} \), then \( \text{NP} = \text{RP} \).

Problem 5-2 \( (x^x = x \text{ for } x \in \{\text{BPP}, \text{RP}\}) \) 10 points
Let \( \text{BPP}^{\text{BPP}} = \bigcup_{B \in \text{BPP}} \text{BPP}^B \). Show that \( \text{BPP}^{\text{BPP}} = \text{BPP} \).
Is \( \text{RP}^{\text{RP}} = \text{RP} \)? Why or why not?

Problem 5-3 (ZPP = RP \( \cap \) coRP) 20 points
Show that \( \text{ZPP} = \text{RP} \cap \text{coRP} \). Remember to argue both directions.
(Hint: Remember Markov’s inequality: if random variable \( X > 0 \), then \( \Pr(X > t\mathbb{E}[X]) \leq 1/t \).)

Problem 5-4 (Logarithmic Advice) 15 points
Recall the class \( \text{P/poly} \) of languages accepted by polynomial time Turing machines having polynomially long (in the input size) advice string. We saw that \( \text{BPP} \subseteq \text{P/poly} \), and mentioned that it is very unlikely that \( \text{NP} \subseteq \text{P/poly} \). Define a more restrictive class \( \text{P/log} \) which is the same as \( \text{P/poly} \) except the advice is only logarithmic in the input size (i.e., there could be only polynomially many advice strings for each input length). Show that \( \text{NP} \subseteq \text{P/log} \) implies that \( \text{P} = \text{NP} \).

Problem 5-5 (Testing Equality) 15 points
Let \( a = a_1 \ldots a_n \), \( b = b_1 \ldots b_n \in \{0, 1\}^n \) be two \( n \)-bit numbers held by Alice and Bob respectively. Alice and Bob try to determine if \( a = b \) using the smallest amount of communication. We mentioned that the deterministic protocol requires \( n \) bits of communication. Here you will compare two probabilistic protocols.

(a) Alice and Bob interpret \( a_i \)'s and \( b_i \)'s as coefficients of degree \( n \) polynomials \( A(x) = \sum_i a_i x^{i-1} \) and \( B(x) = \sum_i b_i x^{i-1} \) over a field \( \mathbb{Z}_p \), where \( p > n^2 \) is a prime. Then Alice picks (after choosing \( p \)) a random \( x \in \mathbb{Z}_p \) and sends \( p, x, A(x) \) to Bob, and Bob agrees if \( A(x) = B(x) \).
This requires \( 6 \log n \) bits of communication. If \( a = b \), then Bob accepts. Assume \( a \neq b \). What is the maximum probability Bob accepts (this is called the error probability)?

(b) Alice picks a random prime \( p \) of length \( s \) bits (to be determined), and sends \( p \) together with \( a \mod p \). Bob agrees if \( a \mod p = b \mod p \). What is the value of \( s \) so that the error probability
of this test is less than in the previous protocol? How does your communication complexity compare with the previous protocol?¹

(Hint: Use the prime number theorem saying that there are approximately \(k/\log k\) primes less than \(k\).)

Problem 5-6 (AM⁰⁰ = AM) 25 points

Recall that AM is a class of languages which have a one round Arthur-Merlin protocol: on input \(x\), the verifier \(V\) sends a random challenge \(R\), the prover \(P\) sends the response \(z\), and the verifier accepts or rejects in probabilistic polynomial time as function of \(x, R, z\). Also, when \(x \in L\), \(\Pr((P \leftrightarrow V)(x) = 1) \geq 2/3\) and when \(x \notin L\), for any prover \(\check{P}\), \(\Pr((\check{P} \leftrightarrow V)(x) = 1) \leq 1/3\).

Now imagine that the verifier does not have access to true randomness, but instead has access to an SV-source with some constant fairness \(\delta > 0\). Namely, the source emits a sequence \(r_1, r_2, \ldots\) where \(\Pr(r_i = 1 \mid r_1 \ldots r_{i-1}) \in (\delta, 1 - \delta)\). We can then define the class \(AM_{sv}\), whose definition is identical to the one of \(AM\) above, except that the completeness and soundness have to hold no matter what particular SV-source the verifier has to deal with. Clearly, \(AM_{sv} \subseteq AM\).

Recall also that in class we showed that for every \(N\), there exists an efficient extractor \(Ext: \{0, 1\}^{O(N)} \times \{0, 1\}^{O(\log N)} \rightarrow \{0, 1\}^N\) for SV sources that extracts \(N\) bits which are 0.1-close to the uniform distribution on \(\{0, 1\}^N\); from any \(O(N)\) semi-random bits and using only \(O(\log N)\)-bit seed.

In this problem you will show that \(AM\) can tolerate SV sources: namely, \(AM \subseteq AM_{sv}\). Specifically, assume \(L \in AM\) is decided by a (unbounded) prover \(P\) and probabilistic polynomial time verifier \(V\). To make your life slightly simpler, assume the completeness of this protocol is 0.99 and soundness is 0.01 (by repeating the original protocol several times in parallel, if needed).

(a) Formally define the new prover \(P_{sv}\) and the new verifier \(V_{sv}\) as a function of \(P, V\) and \(Ext\), which are supposed to be correct for deciding \(L\) using a \(\delta\)-SV source. Make sure you explicitly specify all the parameters of the extractor that you use, and do not forget that \(V\)’s final procedure is also probabilistic.

(b) Formally argue that the resulting protocol is complete (with completeness 2/3), for any particular \(\delta\)-SV-source. Make sure you correctly use Markov’s inequality somewhere.

(c) Formally argue that the resulting protocol is sound (with completeness 1/3), for any particular \(\delta\)-SV-source. Make sure you correctly use Markov’s inequality somewhere.

¹Food for thought. The first protocol fixes the field (as long as it is large enough), and picks a random point \(x\) in the field. In the second protocol, computing \((a \mod p)\) and \((b \mod p)\) can be viewed as evaluating \((A(2) \mod p)\) and \((B(2) \mod p)\) respectively. Incidentally, one can see that the point \(x = 2\) is arbitrary, any point \(x \neq 1\) works as long as it is “small enough” (why?). Thus, the field is now chosen at random, while the point is fixed (as long as it is small).