Problem 1. (Error Reduction for Randomized Algorithms)

1. Let $L$ be a language, and $A$ a randomized algorithm for $L$ such that
   - $x \in L \Rightarrow \Pr_r[A(x; r) \text{ accepts }] \geq 1/p(n),$
   - $x \notin L \Rightarrow \Pr_r[A(x; r) \text{ rejects }] = 1.$

   where $n = |x|$ is the input length, and $p$ is a polynomial. That is, $A$ is an algorithm for $L$ with one-sided error. Let $k(n)$ be the number of random bits used by $A$ (i.e., $|r| = k$), and $t(n)$ be the running time. As mentioned in lecture, we can reduce the error from $1 - p(n)$ to $2^{-q(n)}$ by repetition, where $q(n)$ is a polynomial. Formally, we can define the following algorithm $A'$: On input $x$
   - Run $A$ on $x$ $c(n)$ times.
   - Accept if $A$ ever accepts $x.$

   When $c(n)$ is sufficiently large, we can have
   - $x \in L \Rightarrow \Pr_{r'}[A'(x; r') \text{ accepts }] \geq 1 - 2^{-q(n)},$
   - $x \notin L \Rightarrow \Pr_{r'}[A'(x; r') \text{ rejects }] = 1.$

   Find a sufficiently large $c(n)$, and prove the above statement. What is the number of random bits used by $A'$ and the running time of $A'$?

2. Let $L$ be a language, and $A$ a randomized algorithm for $L$ such that
• $x \in L \Rightarrow \Pr_r[A(x; r) \text{ accepts }] \geq 1/2 + 1/p(n),$
• $x \notin L \Rightarrow \Pr_r[A(x; r) \text{ rejects }] \geq 1/2 + 1/p(n),$

where $n = |x|$ is the input length, and $p$ is a polynomial. That is, $A$ is an algorithm for $L$ with two-sided error. Let $k(n)$ be the number of random bits used by $A$ (i.e., $|r| = k$), and $t(n)$ be the running time. Follow the same argument as Part 1 to reduce the error probability from $1/2 - p(n)$ to $2^{-q(n)}$. (You need to define an algorithm $A''$, and analysis the error probability of $A''$.) What is the number of random bits used by $A''$ and the running time of $A''$?

3. Let $\text{BPP}_s$ be the class of language $L$ such that there exists a probability polynomial-time algorithm $A$ such that $\Pr[A(x) = L(x)] \geq s(|x|)$. Use Part 2 to show that $\text{BPP}_{1/2+p(n)} = \text{BPP} = \text{BPP}_{1-2^{-q(n)}}$ for any polynomials $p(n)$ and $q(n)$.

Remark 1 The point of this problem is to make sure you understand the definition of complexity classes, and the basic error reduction technique. We present the problem and ask you to prove it in a very formal way. Except for this problem, your solution can be less formal. Note the slightly different tradeoff between the number of random bits $k(n)$ and the error probability in Part 1 and 2.

Problem 2. (Randomness in Combinatorics) In this problem, you will see a motivation of derandomization from combinatorics. Let $G$ be a $n$-vertex graph. We say $G$ is a $k$ Ramsey graph if $G$ has no clique or independent set of size $k$. Ramsey graph is a very interesting subject in Combinatorics. We are interested in minimizing $k$. That is, fix $n$, what is the smallest $k$ such that a $n$-vertex $k$ Ramsey graph exists? Following, we will use probabilistic method to prove the existence of Ramsey graphs with $k = 2\log n$. Let $G$ be a $n$-vertex random graph such that each edge $e = (u, v)$ is in $G$ with probability $1/2.$

1. Let $v_1, \ldots, v_k$ be $k$ vertices in $G$, what is the probability that $v_1, \ldots, v_k$ is a clique or an independent set in $G$.

2. Use union bound to give an upper bound of the probability that $G$ has either a clique of size $k$ or an independent set of size $k$.

3. Set $k = 2\log n$, and show that $\Pr_G[G \text{ is not a } k \text{ Ramsey graph}] < 1$. (The following inequality might be useful: $\left(\frac{1}{2}\right)^k \leq \left(\frac{1}{2^k}\right)^k \leq \left(\frac{1}{2^k}\right)^k$, where $e$ is the base of natural log.)

$\Pr_G[G \text{ is not a } 2\log n \text{ Ramsey graph}] < 1$ is equivalent to $\Pr_G[G \text{ is a } 2\log n \text{ Ramsey graph}] > 0$. Therefore, we can conclude that there exists a $2\log n$ Ramsey graph.

The above argument is a typical application of probabilistic method, which is very useful to prove the existence of certain object. We will see more examples later. In fact, we have $\Pr_G[G \text{ is not a } 2\log n \text{ Ramsey graph}] < o(1)$ or $\Pr_G[G \text{ is a } 2\log n \text{ Ramsey graph}] > 1-o(1)$. Therefore, for sufficient large $n$, $\Pr_G[G \text{ is a } 2\log n \text{ Ramsey graph}] > 0.9999$. In other words, most $n$-vertex graph is a $2\log n$ Ramsey graph!

However, it remains open to construct such a graph explicitly (i.e., deterministically, and efficiently). The best known explicit construction achieves $k = 2^{\log^{o(1)} n} > 2\log n$. This is a fairly new (2006) result, which is achieved via the study of another pseudorandom object called disperser.

Note again that we have an efficient randomized construction of $2\log n$ Ramsey graph. (We can simply generate a random graph and output it. The success probability is $> 0.9999$!) The
construction use \( \binom{n}{2} \) random bits. If we can derandomize the construction, or reduce the number of random bits to \( O(\log n) \), then we can get an explicit construction of \( 2\log n \) Ramsey graph (why?).

**Interesting problems**

**Problem 3. (A Different Identity Test)** In this problem, you will analyze an alternative to the identity testing algorithm seen in class, which directly handles polynomials of degree larger than the field size.

The following definitions and facts may be useful: A polynomial \( p(x) \) over a field \( F \) is called **irreducible** if it has no nontrivial factors (i.e. factors other than constants from \( F \) or constant multiples of \( p \)). Analogously to prime factorization of integers, every polynomial over \( F \) can be factored into irreducible polynomials and this factorization is unique (up to reordering and constant multiples). It is known that the number of irreducible polynomials of degree at most \( d \) over a field \( F \) is at least \( \frac{F^d + 1}{2d} \). (This is similar to the Prime Number Theorem for integers, but is much easier to prove.) For polynomials \( p(x) \) and \( q(x) \), \( p(x) \mod q(x) \) is the remainder when \( p \) is divided by \( q \). (More background on polynomials over finite fields can be found in the references in the syllabus.)

In this problem, we consider a version of the identity testing problem where a polynomial \( p(x_1, \ldots, x_n) \) over finite field \( F \) is presented as a formula built up from elements of \( F \) and the variables \( x_1, \ldots, x_n \) using addition, multiplication, and **exponentiation** with exponents given in **binary**. We also assume that we are given a representation of \( F \) enabling addition, multiplication, and division in \( F \) to be done quickly.

1. Let \( p(x) \) be a polynomial of degree \( \leq D \) over a field \( F \). Prove that if \( p(x) \) is nonzero (as a formal polynomial) and \( q(x) \) is a randomly selected polynomial of degree at most \( d = O(\log D) \), then the probability that \( p(x) \mod q(x) \) is nonzero is at least \( \Omega(1/\log D) \). Deduce a randomized, polynomial-time identity test for **univariate** polynomials presented in the above form.

   **Hints:** Consider the following questions: What is a random polynomial of degree at most \( d \)? What can we say if \( q(x) \) is irreducible, and not a factor of \( p(x) \)? What is the probability that \( q(x) \) is irreducible? What is the probability that \( q(x) \) is not a factor of \( p(x) \)?

2. Obtain an identity test for multivariate polynomials by reduction to the univariate case.

   **Hints:** Let \( p(x_1, \ldots, x_n) \) be a multivariate polynomial. Try to convert \( p \) to a univariate polynomial \( p'(y) \) such that \( p \equiv 0 \iff p' \equiv 0 \).

**Problem 4. (A Chernoff Bound)** Let \( X_1, \ldots, X_t \) be independent \([0, 1]\)-valued random variables, and \( X = \sum_{i=1}^{t} X_i \).

1. Show that for every \( r \in [0, 1/2] \), \( E[e^{rX}] \leq e^{rE[X]+r^2t} \). (Hint: \( 1 + x \leq e^x \leq 1 + x + x^2 \) for all \( x \in [0, 1/2] \).)

2. Deduce the following Chernoff Bound: \( Pr[|X - E[X]| + \varepsilon t] \leq e^{-\varepsilon^2 t/4} \). Were did you use the independence of the \( x_i \)'s?
Problem 5. (Robustness of the model) Suppose we modify our model of randomized computation to allow the algorithm to obtain a random element of \(\{1, \ldots, m\}\) for any number \(m\) whose binary representation it has already computed (as opposed to just allowing it access to random bits). Show that this would not change the classes \(\text{BPP}\) and \(\text{RP}\).

hint: Let \(\text{BPP}\) be the class in standard model, and \(\text{BPP}'\) be the class in another model. You need to prove \(\text{BPP} = \text{BPP}'\). That is, for every \(L \in \text{BPP}\), you need to show that \(L \in \text{BPP}'\), and for every \(L \in \text{BPP}'\), you need to show that \(L \in \text{BPP}\).

Problem 6. (Necessity of Randomness for Identity Testing*) In this problem, we consider the “oracle version” of the identity testing problem, where an arbitrary polynomial \(p : \mathbb{F}^m \to \mathbb{F}\) of degree \(d\) is given as an oracle (ie black box) and the problem is to test whether \(p \equiv 0\). Show that any deterministic algorithm that solves this problem when \(m = d = n\) must make at least \(2^n\) queries to the oracle (in contrast to the randomized identity testing algorithm from class, which makes only one query provided that \(|\mathbb{F}| \geq 2n\)).

Is this a proof that \(P \neq \text{RP}\)?