Instructions: Please make sure that your solutions are well-thought-out, complete, concise, rigorous, and clearly written. Each solution sheet should be marked with your name, the course number, the homework number, the problem number and the date. You are allowed to discuss problems and exchange solution ideas with other students, but you must write up your own solutions. If you collaborated with other students on a problem, please acknowledge them in your write up. These are the fun problems to solve. Some of them require a bit of thinking. Please start early!

Problem 1: Product chip testing (5 + 5 + 10 + 10 = 30 Points)

We have seen in the class that the naive algorithm to multiply two $n \times n$ matrices takes $O(n^3)$ time. We have also seen the improvements made to this by Strassen, and the time complexity of Strassen’s matrix multiplication algorithm is $O(n^{\log_2 7})$. However the algorithm is a bit messier. Suppose your friend Binny claims to have designed a chip that can do matrix multiplications. However, you want to make sure that her chip is not faulty, i.e., it does not compute incorrect product of two matrices. Your goal is to quickly verify if her chip computed the product of two matrices correctly.

As usual, you asked your instructor for some help during the office hour. He gave you the following randomized algorithm to check if $A \times B = C$, where $A$, $B$ and $C$ are $n \times n$ matrices:

1. Pick a random vector $r = (r_1, \ldots, r_n)$ such that each $r_i$ is picked independently uniformly at random from a set $S = \{0, 1\}$. (Note: He told you that any finite set $S$ of size $\geq 2$ would also work).

2. If $(AB)r \neq Cr$, then return No; otherwise, return Yes.

In return, he asked you the following questions so that he can assess your progress. Please give answers his questions.

a. Show that the above algorithm runs in $O(n^2)$ time.

b. If we find an $r$ such that $(AB)r \neq Cr$, what is the probability that the algorithm outputs No? If $AB = C$, does the algorithm output Yes for any $r$?

c. Show that if $AB \neq C$, then $\Pr[(AB)r = Cr] \leq \frac{1}{|S|}$.

d. You just proved in (c) that your algorithm has one-sided error. Your instructor told you to repeat the algorithm $k$ times independently to decrease the probability of error. Describe an algorithm to do that. Note that your algorithm invokes the instructor’s algorithm as a subroutine. Give the running time of your algorithm in terms of $n$ and the number of repetitions ($k$). Also give a bound on the probability of error for your algorithm. (Hint: You will see that the probability of error reduces really fast!)

Problem 2: Maximum Satisfiability (4 + 8 + 8 = 20 Points)

In the 3-SAT problem, you are given a set of clauses, where each clause is the disjunction (OR) of three literals (a literal is a Boolean variable or the negation of a Boolean variable). You
are looking for a way to assign a value True or False to each of the variables so that maximum number of clauses are satisfied. A clause is satisfied if at least one of its literals is True. For example, here is an instance of 3-SAT (also called a 3-CNF formula):

\((x_1 \lor \overline{x_2} \lor \overline{x_3}) \land (x_2 \lor x_3 \lor \overline{x_4}) \land (\overline{x_1} \lor x_2 \lor x_4)\).

The assignment \((x_1, x_2, x_3, x_4) = (\text{True}, \text{True}, \text{False}, \text{True})\) satisfies all the clauses. However, the assignment \((x_1, x_2, x_3, x_4) = (\text{False}, \text{True}, \text{True}, \text{False})\) satisfies the second and third clauses.

Given an instance of 3-SAT with \(n\) variables and \(k\) clauses, testing whether all the clauses can be satisfied is known to be a difficult (NP-complete) problem, for which no polynomial-time algorithm is known. In this question, you are asked to design an efficient randomized algorithm to satisfy as many clauses as possible. Your instructor gave you a hint. He gave you a one line randomized algorithm: “set each variable to 0 or 1 with equal probability (1/2).”

a. What is the running time of this algorithm?

b. Show that each clause is not satisfied with a probability of 1/8, from which deduce that the probability that each clause is satisfied is 7/8.

c. Show that the expected number of satisfied clauses is \(7k/8\).

Problem 3: Private Coin

(5 + 5 + 5 + 5 + 5 = 25 Points)

Alice and Bob have infinite computation power. Alice lives on earth, and Bob has gone to Mars to do explorations. Communication (transferring bits) between them is a major bottleneck, and very expensive. Alice has an \(n\)-bit long string \(x\), and Bob has an \(n\)-bit string \(y\). They want to verify if \(x = y\). In the class, we have shown that this can be done with \(n + 1\) bit transfers through the communication channel, in a deterministic setting. We then gave a randomized protocol that uses a shared random string (a.k.a., public coin), shared by both Alice and Bob (i.e., supplied to them beforehand), and showed that only 2 bits need to be transferred (one by Alice and one by Bob) to ensure that the protocol errs (i.e., incorrectly declares \(x = y\), when \(x \neq y\)) with probability at most 1/2.

In this question, we assume that Alice and Bob no longer share the same random string (i.e., no public coin is allowed). But, before Bob left for Mars, he and Alice agreed to use any random prime number \(p\) from the first \(4n\) primes. They used the following protocol to test whether \(x = y\).

1. Alice chooses a random prime \(p\) from the first \(4n\) primes, and then computes \(x \mod p\), and sends Bob the pair \((p, x \mod p)\).

2. Bob computes \(y \mod p\), and checks if it is equal to the value of \(x \mod p\), that Alice sent to him. If they are equal, then Bob transmits 1 to Alice to let her know that the two strings \(x\) and \(y\) are equal; otherwise, he transmits a 0 to Alice.

You are required to answer the following questions.

a. Show that the number of bits transferred between Alice and Bob is \(\Theta(\log n)\). [Hint: What is the size of the largest prime chosen among the first \(4n\) primes?]

b. When \(x = y\), what is the probability that the protocol gives correct answer?

c. How many distinct prime factors (at most) does \(|x - y|\) have?
d. Show that the above protocol errs (i.e., when $x \mod p$ is equal to $y \mod p$, but $x \neq y$ for a bad choice of $p$), with probability at most 1/4.

e. How many times Alice and Bob need to repeat the protocol so that the error probability becomes at most $1/n$? How many bits need to be transferred to ensure this (give asymptotic bound)?

**Problem 4: Binary GCD algorithm**  
(3 + 3 + 3 + 8 + 8 + 5 = 30 Points)
The GCD algorithm we discussed uses division operation. However, division operation can be a costly operation on most computers. On the other hand, subtraction, parity testing, and halving of even numbers (which corresponds to a right shift in the binary representation) can be done efficiently by most computers. This problem requires you to design an algorithm to compute the GCD of two integers without using the division (i.e., remainder) operation.

a. Prove that if $u$ and $v$ are both even, then $\text{gcd}(u, v) = 2 \text{gcd}(u/2, v/2)$.

b. Prove that if $u$ is even and $v$ is odd, then $\text{gcd}(u, v) = \text{gcd}(u/2, v)$.

c. Prove that if $u$ and $v$ are both odd, then $\text{gcd}(u, v) = \text{gcd}(|u - v| / 2, v)$.

d. Use the above ideas to design an efficient iterative binary GCD algorithm for input integers $u$ and $v$, where $u, v > 0$.

e. Give the recursive version of the algorithm.

f. Analyze the time complexity (i.e., asymptotic bound on the number of bit operations) of the above algorithm(s), and show that it is $O((\lg uv)^2)$ bit operations.

**Problem 5: RSA encryption and decryption**  
(10 + 10 = 20 Points)

a. Encrypt the message $M = 0704$ using the RSA cryptosystem using $p = 43$, $q = 59$, $n = pq = 2537$, and $e = 13$. Note that $\text{gcd}(e, (p - 1)(q - 1)) = \text{gcd}(13, 42 \cdot 58) = 1$.

b. Choose a suitable decryption key $d$, and decrypt the above encrypted message.