CS 70 Discrete Mathematics and Probability Theory
Summer 2020 Amin Ghafari, Yining Liu, Khalil Sarwari Midterm

- You may consult two handwritten double-sided sheets of notes. Apart from that, you may not look at books, notes, etc. Calculators, phones, computers, and other electronic devices are prohibited unless they are part of the recording submission. The only tabs/windows you may have open are the Exam PDF, Exam Google Doc, Midterm Instructions Doc and/or Exam Policy (Public) Doc, timer/clock, and Zoom.
- There are 9 questions on this exam, worth a total of 100 points.
- We will not take any clarifying questions; if there is a mistake in the exam, we will resolve it via regrade request or remove the corresponding question/part entirely.
- The questions vary in difficulty, so if you get stuck on any question it may help to leave it and return to it later.
- **Make sure you read the title of the first problem out loud**
- **You may, without proof, use theorems and facts that were proven in the lecture, notes, discussions, and/or in homeworks unless explicitly mentioned otherwise.**
- **You have 120 minutes to work on the exam. After this time, you may no longer work on the exam. You will then have 30 minutes for scanning and uploading your answers. Late submissions will be penalized.**

1 (10 points) Logic Questions

**Before starting this question, please read the title out loud: “Logic Questions”.
This is to verify it is not a prerecorded video. Not doing this will void your exam.**

(a) (3 points) **First** write each of the following two statements in the form “$p \implies q$” in English. **Then** state either the converse or contrapositive of each of the statements in English as well.
For example: for the statement “I get up early on weekdays”, the statement in English is “weekdays $\implies$ get up early”. For each of your six answers, you may use the implication symbol $\implies$, but should otherwise use only English:

(i) I always listen to music when I am happy.

Statement:

Contrapositive:

(ii) I will answer your question only if you make a Piazza post.

Statement:

Converse:

Answer:

- I always listen to music when I am happy.
  Statement: I am happy $\implies$ I am listening to music.
  Contrapositive: I am not listening to music $\implies$ I am not happy.
- I will answer your question only if you make a piazza post.
  Statement: I answer your question. $\implies$ You make a Piazza post
  Converse: You make a Piazza post $\implies$ I answer your question.

(b) (2 points) Let $P(x)$ be the statement $x = x^2$. If the domain consists of integers, what are the truth values of the proposition $P(0)$ and the proposition $\forall x\, P(x)$?

Answer: The truth value for $P(0)$ is True because $0 = 0^2$. The truth value of $\forall x P(x)$ is False because, for example, $2 \neq 2^2$.

(c) (2 points) Let $B(x)$ be the statement "x likes boba" and $J(x)$ be the statement "x can program in Java" and $F(x,\ y)$ be the statement "x and y are friends". The domain of quantifiers consists of all students in CS70 Summer 2020. Express the following sentence in terms of $B(x)$, $J(x), F(x,\ y)$, quantifiers, and logical connectives.

"Every student has a friend in CS70 who likes boba but doesn’t know how to program in Java."

Answer: $\forall x \exists y (F(x, y) \wedge (\neg J(y) \wedge B(y)))$

(d) (3 points) Prove $(\neg Q \wedge (P \implies Q)) \implies \neg P$ is a tautology using logical equivalences (i.e. answers that use truth tables or explaining in words will get zero points).

Answer: Yes.

$$\begin{aligned}
(\neg Q \wedge (P \implies Q)) \implies \neg P &\equiv \neg(\neg Q \wedge (\neg P \vee Q)) \vee \neg P \\
&\equiv \neg((\neg Q \wedge \neg P) \vee (\neg Q \wedge Q)) \vee \neg P \\
&\equiv \neg((\neg Q \wedge \neg P) \vee \mathbf{F}) \vee \neg P \\
&\equiv \neg(\neg Q \wedge \neg P) \vee \neg P \\
&\equiv (Q \vee P) \vee \neg P \\
&\equiv Q \vee (P \vee \neg P) \\
&\equiv Q \vee \mathbf{T} \\
&\equiv \mathbf{T}
\end{aligned}$$

2 (16 points) Proofs

(a) (3 points) The following is a false proof. **Locate** (state which step is wrong) and **explain** the error.

Proposition: Given $n \in \mathbb{N}$, for any set S with $|S| = n$, then S must be a set of n zeros.

Proof:

Step 1: If $n = 0$, S is the empty set, thus the statement is trivially true.

Step 2: Let $k \in \mathbb{N}$. Assume the statement is true for a set S whenever $|S| \leq k$.

Step 3: Now consider a set S where $|S| = k+1$.

Step 4: Divide $S = S_1 \cup S_2$ such that $S_1 \cap S_2 = \emptyset$ and $0 < |S_1|, |S_2| < k+1$.

Step 5: Apply the inductive hypothesis on S_1, S_2, we see that S_1, S_2 contain only zeros. Since $S = S_1 \cup S_2$, we know S also only contain zeros.

Answer: Step 4 is wrong. Recall that to prove a proposition "$\forall n \in \mathbb{N}, P(n)$" using induction, you need to show 1) $P(0)$ holds, and 2) for all $n \in \mathbb{N}, P(n) \implies P(n+1)$.

If $|S| = 1$, step 4 doesn't work, since we are not able to divide such S into disjoint subsets S_1, S_2 such that $0 < |S_1|, |S_2| < 1$. As a result, $P(0) \not\Longrightarrow P(1)$ for this proposition.

(This example is similar to the "fun problem" at the beginning of Lecture 3 and the "All horses are the same color" false proof in Note 3.)

(b) (5 points) Fibonacci sequence is defined such that each number is the sum of the two preceding ones, starting from 0 and 1. That is

$$F_0 = 0, \quad F_1 = 1, \qquad F_n = F_{n-1} + F_{n-2}.$$

Prove by induction that $F_n^2 - F_{n+1}F_{n-1} = (-1)^{n-1}$.

(HINT: You may need to use the property $F_n = F_{n-1} + F_{n-2}$ a couple of times in your inductive step of the proof. Your proof shouldn't be very long!)

Answer:

- **Base Case:** $n = 1$, $P(1)$: $F_1^2 - F_2F_0 = 1 - 1 \times 0 = 1 = (-1)^0 = (-1)^{1-1}$.
- **Inductive Hypothesis:** Assume for some arbitrary $k \geq 1$ we have $P(n = k) : F_k^2 - F_{k+1}F_{k-1} = (-1)^{k-1}$.
- **Inductive Step:** We show this is correct for $k+1$, i.e. $P(k+1) : F_{k+1}^2 - F_{k+2}F_k = (-1)^k$.

$$\begin{aligned}
F_{k+1}^2 - F_{k+2}F_k =& F_{k+1}^2 - (F_{k+1} + F_k)F_k && \text{From Fibonacci sequence: } F_{k+2} = F_{k+1} + F_k \\
=& F_{k+1}^2 - F_{k+1}F_k - F_k^2 \\
=& F_{k+1}(F_{k+1} - F_k) - F_k^2 \\
=& F_{k+1}(F_{k-1}) - F_k^2 && \text{From Fibonacci sequence: } F_{k-1} = F_{k+1} - F_k \\
=& -(F_k^2 - F_{k+1}F_{k-1}) \\
=& -(-1)^{k-1} && \text{From Inductive Hypothesis.} \\
=& (-1)^k.
\end{aligned}$$

So $P(k+1) : F_{k+1}^2 - F_{k+2}F_k = (-1)^k$. Hence, by the induction principle $P(n) : F_n^2 - F_{n+1}F_{n-1} = (-1)^{n-1}$ for any $n \geq 1$.

(c) (5 points) Let $f : X \to Y$. Recall that for any $S \subseteq X$, $f(S) = \{f(s) : s \in S\}$.

(i) Given $A, B \subseteq X$, prove that $f(A) \setminus f(B) \subseteq f(A \setminus B)$.

(ii) Give an example of A, B and $f : X \to Y$ such that $f(A) \setminus f(B) \neq f(A \setminus B)$

Answer: Consider any $y \in f(A) \setminus f(B)$. That is, $y \in f(A)$ and $y \notin f(B)$. Hence, there exists $a \in A$ such that $y = f(a)$ and $a \notin B$. Since $a \in A \setminus B$, we have $y = f(a) \in f(A \setminus B)$. Thus, $f(A) \setminus f(B) \subseteq f(A \setminus B)$.

An example is $f : \{0,1\} \to \{2\}$ where f maps both 0 and 1 to 2. Consider $A = \{0,1\}$ and $B = \{0\}$. Then we have $f(A) \setminus f(B) = \{2\} \setminus \{2\} = \emptyset$ whereas $f(A \setminus B) = f(\{1\}) = \{2\}$.

(d) (3 points) Prove that there are no solutions in integers x and y to the equation $x^2 - 3y^3 = 2$. *(HINT: Consider the equation modulo 3.)*

Answer: Suppose there exist $x, y \in \mathbb{Z}$ such that $x^2 - 3y^3 = 2$.

Then we must have $x^2 - 3y^3 \equiv 2 \pmod 3$, which means $x^2 \equiv 2 \pmod 3$. Since $0^2 \equiv 0 \pmod 3$, and $1^2 \equiv 2^2 \equiv 1 \pmod 3$, such x doesn't exist.

3 (8 points) Tiny Sets

Given $X \subseteq \mathbb{R}$, we say X is *tiny* if for each $\varepsilon > 0$, we can find a countable collection of closed intervals $[a_1, b_1], [a_2, b_2], \dots$ such that

(i) $X \subseteq \bigcup_{n=1}^{\infty} [a_n, b_n]$;

(ii) $\sum_{n=1}^{\infty} (b_n - a_n) < \varepsilon$.

Intuitively, X is tiny if we can find an arbitrarily small cover of X using closed intervals.

(a) (2 points) Prove that any singleton set, i.e. $\{x\}$ where $x \in \mathbb{R}$, is tiny.

Answer: We can use $[x,x],[x,x]\dots$ as the cover, because $\{x\} \subseteq \cup_{n=1}^{\infty}[x,x]$ and $\sum_{n=1}^{\infty}(x-x) = 0 < \varepsilon$.

Alternatively, a finite collection of closed intervals is a countable collection of closed intervals. Given ε, we can use $[x - \frac{\varepsilon}{3}, x + \frac{\varepsilon}{3}]$ to cover $\{x\}$, i.e. $\{x\} \subset [x - \frac{\varepsilon}{3}, x + \frac{\varepsilon}{3}]$. Furthermore, $(x + \frac{\varepsilon}{3}) - (x - \frac{\varepsilon}{3}) = \frac{2}{3}\varepsilon < \varepsilon$, so the cover is sufficiently small. The above argument holds for all $x \in \mathbb{R}$ and $\varepsilon > 0$, showing all singleton sets are tiny.

(b) (i) (4 points) Let $A_1, A_2, \dots$ be a countable collection of tiny sets. Prove $\cup_{n=1}^{\infty} A_n$ is tiny. *(HINT: you may find $\sum_{n=1}^{\infty} \frac{1}{2^n} = 1$ to be helpful.)*

(ii) (2 points) Is $\mathbb{Q}$ tiny? Justify your answer.

Answer:

(i) Let $\varepsilon > 0$. For each A_n, we can find a countable collection of intervals $[a_{n_1}, b_{n_1}], [a_{n_2}, b_{n_2}], \dots$ such that

- $A_n \subseteq \bigcup_{k=1}^{\infty} [a_{n_k}, b_{n_k}]$;
- $\sum_{k=1}^{\infty} (b_{n_k} - a_{n_k}) < \frac{\varepsilon}{2^n}$.

In Discussion 2A, we've seen that $\mathbb{N} \times \mathbb{N}$ is countable. Therefore, $[a_{n_k}, b_{n_k}]$ where $n, k \in \mathbb{N}$ form a countable collection of intervals such that

$$\bigcup_{n=1}^{\infty} A_n \subseteq \bigcup_{n,k \in \mathbb{N}} [a_{n_k}, b_{n_k}].$$

To see the cover is also small, use the hint, and we get

$$\sum_{n,k \in \mathbb{N}} (b_{nk} - a_{nk}) = \sum_{n=1}^{\infty} \sum_{k=1}^{\infty} (b_{nk} - a_{nk}) \le \sum_{n=1}^{\infty} \frac{\varepsilon}{2^n} = \varepsilon \sum_{n=1}^{\infty} \frac{1}{2^n} = \varepsilon$$

(ii) Yes. We've seen in Lecture 5 that $\mathbb{Q}$ is countable, i.e. $\mathbb{Q}$ is a countable union of singletons. By (a), singletons are tiny. Therefore $\mathbb{Q}$ is tiny by (b).

4 (8 points) Computability

Let P and Q be programs (finite length bit-strings), and x is an input (finite length bit-string). Consider the program $FASTER(P, Q, x)$ which returns True if $P(x)$ takes strictly fewer steps than $Q(x)$ to execute, and returns False otherwise. Note: If $P(x)$ and $Q(x)$ both take infinitely many steps, then $FASTER$ returns False.

(a) (3 points) Is $FASTER$ computable? If so, provide pseudocode for the program $FASTER$. If not, prove that it is uncomputable.

Answer: No, it is not computable. Let $LOOPFOREVER$ be some program that loops forever. The $TestHalt$ program reduces to $FASTER$ as follows:

$$TestHalt(P, x) = \begin{cases} True, & \text{if } FASTER(P, LOOPFOREVER, x) \text{ returns True} \\ False, & \text{otherwise.} \end{cases}$$

If P loops forever, $FASTER(P, LOOPFOREVER, x)$ will return False, as reiterated in the “Note:" in this problem.
If P does not loop forever, then it must take strictly fewer steps than $LOOPFOREVER$,
so $FASTER(P, LOOPFOREVER, x)$ will return True.
This is precisely the behavior of TestHalt, which we know is uncomputable. Since the computability of $FASTER$ implies the computability of $TestHalt$, by contraposition we know that $FASTER$ must be uncomputable.

(b) (5 points) Construct a program $ALLFASTER(x)$ which prints out all tuples (P, Q) satisfying the condition where $P(x)$ takes strictly fewer steps than $Q(x)$ to execute. For every pair of programs P, Q where $P(x)$ terminates before $Q(x)$ the tuple (P, Q) must be printed out after a finite number of steps. Tuples may be printed out multiple times so long as they satisfy the condition.

Answer: $ALLFASTER$ must consider all P and Q without getting stuck simulating a particular program indefinitely. This means when we simulate, we must do so for a fixed number of steps. As we learned in HW2 Q4 Part (a), one viable way to do this simulation is to use dovetailing; namely, we zigzag through programs, inputs, and steps such that we eventually reach every tuple at some finite index.

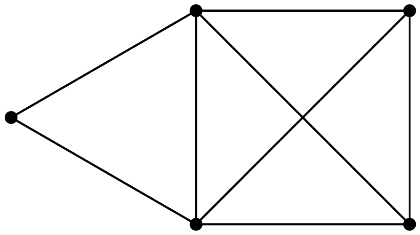
Recall that programs correspond to finite length bit strings, and that we can enumerate these sets. In particular, we can enumerate the set of all programs $P = \{0,1\}^*$, where P_i is the ith element of the enumeration. We can also do this to get a set of all programs Q. We can do these enumerations since the set of programs (finite length bitstrings) is countably infinite. Since we can do this, we can define $ALLFASTER$ as follows:

```
$ALLFASTER(x)$:
  For $k = 1, \dots, \infty$
    For $i = 1, \dots, k$
      For $j = 1, \dots, k$
        If $P_i(x)$ terminates before $Q_j(x)$ within these $k$ steps
          print $(P_i, Q_j)$
```

5 (12 points) Chromatic Numbers

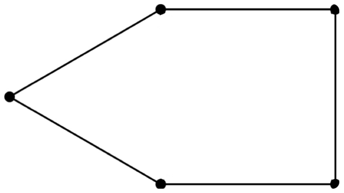
The *chromatic number* of a graph $G = (V, E)$, written $\chi(G)$, is the minimum number of colors required to assigned a color to each vertex of G such that no two adjacent vertices are assigned the same color.

Now, define the *clique number* of a graph $G = (V, E)$, written $\omega(G)$, to be the number of vertices in the largest complete subgraph of G.

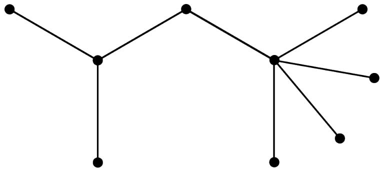


$\chi(G) = 4$ and $\omega(G) = 4$ for the graph G above.

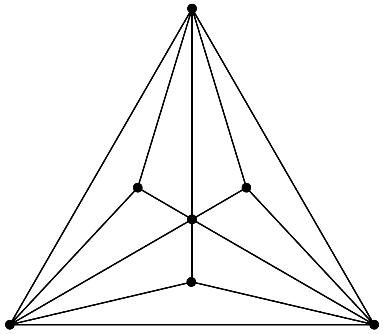
(a) (3 points) Determine the chromatic number and the clique number of each of the following graphs.



(i) $\chi(G_1) =$ $\omega(G_1) =$



(ii) $\chi(G_2) =$ $\omega(G_2) =$



(iii) $\chi(G_3) =$ $\omega(G_3) =$

Answer: The chromatic numbers and clique numbers are determined by inspection. We can determine the chromatic number of G_3 by making a greedy color assignment (using a new color only when necessary). We can determine the clique number of G_3 by noting that the only vertex connected to more than 3 vertices is the central vertex, but we see that it does not form K_5 or K_6, so $\omega(G_3) \leq 4$. We then observe that the three three corner vertices and the central vertex are all mutually connected.

(i) $\chi(G_1) = 3$, $\omega(G_1) = 2$

(ii) $\chi(G_2) = 2$, $\omega(G_2) = 2$

(iii) $\chi(G_3) = 4$, $\omega(G_3) = 4$

(b) (5 points) Let K_n be the complete graph on n vertices. Determine an expression for $\chi(K_n)$ in terms of only n as a variable. Prove your result.

Answer: By inspection and examples, we will prove that $\chi(K_n) = n$.

We see that $\chi(K_n) \leq n$, since coloring every vertex a different color will always be a proper vertex coloring. If we can show that $\chi(K_n) \not< n$, then it must be the case that $\chi(K_n) = n$, as desired.

Suppose for contradiction that $\chi(K_n) < n$. Then there is a proper coloring of K_n using $n-1$ colors. By the pigeonhole principle, at least two vertices of K_n are the same color, since there are n vertices and

$n-1$ colors; call these vertices u and v. Since all vertices in K_n are adjacent, u and v are adjacent. The vertices u and v are adjacent but have the same color; this contradicts the fact that we have a proper coloring of K_n. Therefore $\chi(K_n) \not< n$. Then it must be the case that $\chi(K_n) = n$.

(c) (4 points) Prove that for any graph G, $\chi(G) \geq \omega(G)$.

Answer: We prove that for any subgraph H of G, $\chi(H) \leq \chi(G)$. Suppose H is an arbitrary subgraph of G. Then any proper coloring of G gives a proper coloring of H, since each edge in H is also in G and any pair of adjacent vertices in H are also adjacent in G; therefore, any pair of adjacent vertices in H must be different colors in any proper coloring of G. Therefore $\chi(H) \leq \chi(G)$.

Now, let H be the largest complete subgraph of G. Then, by definition, H has $\omega(G)$ vertices and $\chi(H) = \omega(G)$ by part (b). From the earlier result, $\omega(G) = \chi(H) \leq \chi(G)$, so $\chi(G) \geq \omega(G)$.

6 (17 points) Cicada Party

There are two types of cicadas: type A comes out every 13 years (so A comes out on year $0, 13, 26, \ldots$), and type B comes out every 17 years, (so B comes out on year $0, 17, 34, \ldots$). Assume that it is currently year 0, and both types of cicadas come out this year. You will have to do some calculation in this question.

It may be useful to know: $13^{-1} \equiv 4 \pmod{17}$ and $17^{-1} \equiv 10 \pmod{13}$.

(a) (3 points) What is the next year that both types of cicadas come out?

Answer: 221. This is the least common multiple. Since 13 and 17 are co-prime, then we can calculate the least common multiple by just multiplying them together.

(b) (5 points) A "so close" year is a year when one type of cicada comes out, but the other type of cicada came out the year before! For example, year 170 will be a "so close" year because B comes out in year 170 but A had already came out in 169.

What is the first "so close" year?

Answer: 52.

Let's calculate the (first) year, x, when A comes out and B came out one year before. In this case, we get that $x \equiv 0 \pmod{13}$ and $x \equiv 1 \pmod{17}$, which is satisfied by $13 \times (13^{-1} \mod 17) = 13 \times 4 = 52$.

For the first year x, where B comes out and A come out one year before, we have $x \equiv 0 \pmod{17}$ and $x \equiv 1 \pmod{13}$, which is satisfied by $17 \times (17^{-1} \mod 13) = 17 \times 10 = 170$.

The smaller of the two is 52.

(c) (4 points) The cicadas celebrate a "lucky" year when A is 3 years in its cycle (e.g. year $3, 16 \ldots$) and B is 5 years in its cycle (e.g. year $5, 22, \ldots$).

What is the first "lucky" year?

Answer: We want to find the smallest positive integer solution to

$$x \equiv 3 \pmod{13}$$

$$x \equiv 5 \pmod{17}$$

From Chinese Remainder Theorem, the unique solution mod 221 is:

$$\begin{aligned} x &= 3 \times 17 \times (17^{-1} \mod 13) + 5 \times 13 \times (13^{-1} \mod 17) \mod 221 && (1) \\ &= (3 \times 170 + 5 \times 52) \mod 221 && (2) \\ &= 107 && (3) \end{aligned}$$

where in step (2) we can use our work from part (b). So, 107 is smallest positive integer solution, and therefore the first "lucky" year.

(d) (5 points) It is now year 2020. How many "lucky" years have occurred since year 0?

Answer: 9. Recall the uniqueness part of the Chinese Remainder Theorem says $107 + 221n$ for $n \in \mathbb{Z}$ would be all solutions to the system of linear congruences from (c). We want to find the largest $n \geq 0$ such that $107 + 221n \leq 2020$. Hence, $107 + 221n$ for $n = 0, 1, 2, \ldots, 8$ are all "lucky" years by 2020, so there are 9 "lucky" years.

7 (12 points) Key Exchange

Given a prime p, we say an integer $1 \leq g \leq p-1$ is a *primitive root* if for any $a \in \{1,\dots,p-1\}$, there exists a unique $k \in \{1,\dots,p-1\}$ such that $g^k \equiv a \pmod{p}$. In other words, exponentiation of g would give us all the elements in $\{1,\dots,p-1\}$. It can be proven that a primitive root exists given any prime p.

(a) (1 point) Is the following statement correct? "1 is not a primitive root for all prime $p > 2$."

Briefly justify your answer.

Answer: The statement is correct, since for all k, $1^k \equiv 1 \pmod{p}$ and specifically $1^k \not\equiv 2 \pmod{p}$ and $2 \in \{1,\dots p-1\}$ for all $p > 2$.

Alice and Bob want to agree upon a key that they will use for future communications. They decide to use the following scheme:

Setup: They pick a prime p, a primitive root g and make (p,g) public. Alice and Bob each picks an integer between 1 and $p-1$ as their own private keys, i.e. Alice picks $s_A \in \{1,\dots,p-1\}$ and Bobs picks $s_B \in \{1,\dots,p-1\}$ without sharing the private keys with each other.

Protocol:

1. First, Alice computes $m_A = g^{s_A} \mod p$ and sends it to Bob, and Bob computes $m_B = g^{s_B} \mod p$ and sends it to Alice. Eve **can** see m_A and m_B.
2. Next, Alice computes $k_A = m_B^{s_A} \mod p$ and Bob computes $k_B = m_A^{s_B} \mod p$.

Note that Eve **cannot** see k_A or k_B since those are never shared over any network.

(b) (2 points) Recall that the goal is to agree upon a key for future communications. Show that $k_A = k_B$.

Answer: $k_A \equiv m_B^{s_A} \equiv (g^{s_B})^{s_A} \equiv g^{(s_B \cdot s_A)} \equiv g^{(s_A \cdot s_B)} \equiv (g^{s_A})^{s_B} \equiv m_A^{s_B} \equiv k_B \pmod{p}$.

Since $0 \leq k_A, k_B < p$, we have $k_A = k_B$.

(c) (3 points) The *discrete logarithm problem* asks: given a prime p, a primitive root g, and an integer $1 \leq a \leq p-1$, find the $k \in \{1,\dots,p-1\}$ such that $g^k \equiv a \pmod{p}$.

Show that if Eve can efficiently solve the discrete logarithm problem, she will be able to efficiently compute k_A.

Answer: Since Eve sees p, g, m_A and m_B, she can use discrete log to efficiently compute s_A from $m_A = g^{s_A} \mod p$. Now, she can compute $(m_B)^{s_A} \mod p$, which is k_A.

(d) (4 points) [This question is void due to the information given in the problem, please see the solutions.] Show that if Eve can solve the discrete logarithm problem efficiently, then she can also break RSA efficiently.

(HINT: Recall that breaking RSA means figuring out the private key d given only the public key (N,e), where N is the product of two large primes.)

Answer: NOTE: This question is flawed. In the previous part we defined the Discrete Logarithm Problem (DLP) **only for prime moduli**. This does NOT help us break RSA, since we need to find the discrete logarithm with a composite modulus. Thus we decided to give everyone points for this part.

However, if we assume that we can solve DLP with a composite modulus, we *can* break RSA (this is what we had intended). The next paragraph explains how.

Let (N,e) be the RSA public key, where $N = pq$ for some hidden large primes p and q, and $d = e^{-1} \bmod (p-1)(q-1)$ be the decryption key (the private key). Now, Eve can choose any message m, encrypt it to get $x = m^e \pmod{N}$. Next, she will try to decrypt it. Remember that $x^d \equiv m \pmod{N}$, hence she can use the discrete logarithm to find d, the power to which x must be raised to recover m.

(e) (2 points) Evil Eve was able to bribe Bob to give her s_B. Given knowledge of s_B, can Eve decrypt the messages that Alice and Bob are sharing? Keep in mind that Eve can see all the encrypted communications between Alice and Bob. (Eve cannot take the discrete logarithm!)

Answer: Yes, she can compute $k_A = k_B = m_A^{s_B} \mod p$ and decrypt the communication between Alice and Bob.

8 (12 points) GCD for Polynomials

A common divisor of $P(x)$ and $Q(x)$ is a polynomial $D(x)$ that divides $P(x)$ and $Q(x)$, i.e. $P(x) = D(x)P'(x)$ and $Q(x) = D(x)Q'(x)$ for some polynomials $P'(x)$ and $Q'(x)$. Furthermore, $D(x)$ is the greatest common divisor (GCD) of $P(x)$ and $Q(x)$, if every common divisor of $P(x)$ and $Q(x)$ also divides $D(x)$.

We can use the Euclidean algorithm to find the GCD of any pair of polynomials. It makes repeated use of Euclidean division. When using this algorithm on two numbers, the size of the numbers decreases at each stage. With polynomials, the degree of the polynomials decreases at each stage. The last nonzero remainder, made monic (i.e. it has 1 as coefficient of the highest degree) if necessary, is the GCD of the two polynomials.

Assume we wish to find $\mathrm{GCD}(P(x), Q(x))$ where we know:

$$\deg(Q(x)) \leq \deg(P(x))$$

We can find polynomials $A(x)$ and $B(x)$ such that

$$P(x) = A(x)Q(x) + B(x), \qquad \deg(B(x)) < \deg(Q(x)).$$

Thus, for the $\mathrm{GCD}(P_0 = P(x), Q_0 = Q(x))$ we have

$$\begin{aligned} D(x) &= \mathrm{GCD}(P_0 = P(x), Q_0 = Q(x)) \\ &= \mathrm{GCD}\Big(Q(x), P(x) \pmod{Q(x)}\Big) \\ &= \mathrm{GCD}\big(Q(x), B(x)\big) = \mathrm{GCD}(P_1, Q_1) \end{aligned}$$

We can repeat this for i steps:

$$D(x) = \mathrm{GCD}(P_0, Q_0) = \mathrm{GCD}(P_1, Q_1) = \cdots = \mathrm{GCD}(P_i, Q_i)$$

until $Q_{i=N}(x) = 0$, and the GCD is

$$D(x) = \mathrm{GCD}(P_0, Q_0) = \mathrm{GCD}(P_1, Q_1) = \cdots = \mathrm{GCD}(P_N, 0) = P_N$$

(a) (3 points) Use this algorithm to find the GCD of $P(x) = x^3 + x^2 + x + 1$ and $Q(x) = x^2 + x$.

Answer: Following the Euclid algorithm we have

$$\begin{aligned} \mathrm{GCD}(x^3 + x^2 + x + 1, x^2 + x) &= \mathrm{GCD}(x^2 + x, x + 1) \\ &= \mathrm{GCD}(x + 1, 0) \\ &= x + 1, \end{aligned}$$

So $\mathrm{GCD}(P(x), Q(x)) = x + 1$.

(b) (4 points) We say two polynomials are *coprime* if they share no common polynomial factor, i.e. $P(x)$ being $Q(x)$ coprime means

$$\mathrm{GCD}\big(P(x), Q(x)\big) = 1.$$

Show that if $P(x)$ and $Q(x)$ are coprime then the multiplicative inverse of $P(x)$ mod $Q(x)$ exists.

(HINT: Bezout's theorem for polynomials states that if GCD$\big(P(x), Q(x)\big) = D(x)$, then there exist polynomials $A(x)$ and $B(x)$ such that $D(x) = A(x)P(x) + B(x)Q(x)$.)

Answer: Since GCD$\big(P(x), Q(x)\big) = 1$, we can find polynomials $A(x)$ and $B(x)$ such that

$$A(x)P(x) + B(x)Q(x) = 1.$$

Taking $\pmod{Q(X)}$, we get

$$\begin{aligned} A(x)P(x) + B(x)Q(x) &\equiv 1 \pmod{Q(x)} \\ A(x)P(x) + 0 &\equiv 1 \pmod{Q(x)} \\ A(x)P(x) &\equiv 1 \pmod{Q(x)} \end{aligned}$$

Hence, the multiplicative inverse of $P(x)$ mod $Q(x)$ exists and it is $A(x)$.

(c) (5 points) Show that $P(x) = x^3 + x^2 + 1$ and $Q(x) = x^2 + 1$ are coprime. Then, find the multiplicative inverse of $P(x)$ mod $Q(x)$.

Answer: Following the Euclid algorithm we have

$$\begin{aligned} \mathrm{GCD}(x^3 + x^2 + 1, x^2 + 1) &= \mathrm{GCD}(x^2 + 1, -x) \\ &= \mathrm{GCD}(-x, 1) \\ &= \mathrm{GCD}(1, 0) \\ &= 1. \end{aligned}$$

So $P(x)$ and $Q(x)$ are coprime.
For the multiplicative inverse we have:

$$\begin{aligned} 1 &= 1 \times (x^2 + 1) + x \times (-x) \\ -x &= 1 \times (x^3 + x^2 + 1) + (-x - 1) \times (x^2 + 1). \end{aligned}$$

We substitute from the second in equation in to the first one

$$\begin{aligned} 1 &= 1 \times (x^2 + 1) + x \times [1 \times (x^3 + x^2 + 1) + (-x - 1) \times (x^2 + 1)] \\ \Rightarrow 1 &= (1 - x - x^2) \times (x^2 + 1) + x \times (x^3 + x^2 + 1) \end{aligned}$$

Taking $\pmod{x^2 + 1}$ we have

$$\begin{aligned} 1 &\equiv (1 - x - x^2) \times (x^2 + 1) + x \times (x^3 + x^2 + 1) \pmod{x^2 + 1} \\ \Rightarrow 1 &\equiv x \times (x^3 + x^2 + 1) \pmod{x^2 + 1} \end{aligned}$$

So

$$(x^3 + x^2 + 1)^{-1} \equiv x \pmod{x^2 + 1}$$

Alternative Solution:

$$\begin{tikzcd}
\mathrm{egcd}(x^3+x^2+1, x^2+1) \arrow[r] & \mathrm{egcd}(x^2+1, -x) \arrow[r] & \mathrm{egcd}(-x, 1) \arrow[r] & \mathrm{egcd}(1, 0) \\
(1, x, -x^2-x+1) & (1, 1, x) \arrow[l] & (1, 0, 1) \arrow[l] & (1, 1, 0) \arrow[l]
\end{tikzcd}$$

So we have

$$x(x^3+x^2+1)+(-x^2-x+1)(x^2+1)=1$$

Taking $\pmod{x^2+1}$ we have

$$\Rightarrow x(x^3+x^2+1) \equiv 1 \pmod{x^2+1}.$$

9 (5 points) Degrading Channels

Alice would like to send a secure message to Bob over a channel of size n. This means Alice can send at most n packets at a time across the channel. However, the channel is not very reliable.

Assume the channel behaves as follows: Of the first batch of n packets, it corrupts none; of the second batch of n packets, it corrupts exactly 1; of the third batch of n, it corrupts exactly 2; and so on, until for the $(n+1)^{th}$ batch of n packets (and thereafter), it corrupts all of them.

Suppose we use error correcting codes for each batch of packets in order to recover the original messages which is sent through the channel. What is the maximum size message (in terms of packets) that we can send? **Justify your answer. Assume n is even.**

Your final answer should be a closed-form expression (not a summation).

Answer: In a batch with k corruptions, the maximum number of packets, m_k, we can send using error correcting codes must satisfy $m_k + 2k = n$ or $m_k = n - 2k$. The total number of packets is $\sum_{k=0}^{n/2} m_k = \sum_{k=0}^{n/2} n - 2k$.

Or assuming n is even then the number of packets in the message is at most

$$\begin{aligned}
\sum_{k=0}^{n/2} m_k = \sum_{k=0}^{n/2}(n-2k) &= n + (n-2) + (n-4) + \ldots + 2 \\
&= 2\left(\frac{n}{2} + \frac{n-2}{2} + \frac{n-4}{2} + \ldots + 1\right) \\
&= 2\left(\frac{n}{2} + \left(\frac{n}{2} - 1\right) + \left(\frac{n}{2} - 2\right) + \ldots + 1\right) \\
&= 2\frac{\frac{n}{2}\left(\frac{n}{2}+1\right)}{2} \\
&= \frac{n^2}{4} + \frac{n}{2}.
\end{aligned}$$

Submission

- Keep the recording going;
- Scan answer booklet and cheatsheets into PDF;
- Submit to Gradescope by 10:30PM PDT. If Gradescope is being really slow, you may submit your exam PDF using this form: (emergency only);
- State out loud *"I, [name], finished the exam entirely on my own, and submitted the exam to Gradescope at [time]."*;
- Stop the recording;
- Upload recording to your Google Drive;
- Submit Google Drive link to your uploaded recording using this form:
  by 11:59PM 7/14 PDT.
- Submit your cheatsheets to Gradescope under "Midterm Cheat Sheets". If you did not use any cheatsheets, you must submit 4 blank pages.

If you have technical issues during the exam, you should report these issues when you submit your exam by emailing su20@eecs70.org.

Contributors:

- Amin Ghafari.
- Yining Liu.
- Khalil Sarwari.
- Sagnick Bhattacharya.
- Shahzar Rizvi.
- Kevin Zhu.
- Robert Wang.
- Alan He.