MATH 470: Communications and Cryptography

Homework 2

Due date: 7 September 2023 *Name: Huy Lai*

Problem 1. For each of the $\gcd(a,b)$ values in Exercise 1.9, use the extended Euclidean algorithm (Theorem 1.11) to find integers u and v such that $au + bv = \gcd(a,b)$. Let $a = 291$ and $b = 252$

Solution:

q	r	u	v
	291	1	0
	252	0	1
1	39	1	-1
6	18	-6	7
2	3	13	-15
	0	u	v

Table 1: Extended Euclidean Algorithm

$$u = 13, v = -15$$

Problem 2. Let $a_1, a_2, \ldots, a_k$ be integers with $\gcd(a_1, a_2, \cdots, a_k) = 1$ i.e., the largest positive integer dividing all of $a_1, a_2, \ldots, a_k$ is 1. Prove that the equation

$$a_1u_1 + a_2u_2 + \cdots + a_ku_k = 1$$

has a solution in integers $u_1, u_2, \ldots, u_k$. (*Hint.* Repeatedly apply the extended Euclidean algorithm, Theorem 1.11. You may find it easier to prove a more general statement in which $\gcd(a_1, \ldots, a_k)$ is allowed to be larger than 1.)

Solution:
Proof by Induction

Proof. Base case: $k = 1$
$a_1 \cdot 1 = \gcd(a_1)$

$k = 2$
This is proven by the Extended Euclidean Algorithm

Inductive Hypothesis: Assume $\forall k \geq 3$ that

$$a_1u_1 + a_2u_2 + \cdots + a_{k-1}u_{k-1} = \gcd(a_1, a_2, \cdots, a_{k_1})$$

Let $b = \gcd(a_1, \cdots, a_{k-1})$.
Apply the Extended Euclidean algorithm to b and a_k, this gives the solution

$$bu + a_kv = \gcd(b, a_k)$$

Multiplying the Inductive Hypothesis by v results in

$$\begin{aligned} a_1u_1v + a_2u_2v + \cdots + a_{k-1}u_{k-1}v &= \gcd(a_1, a_2, \cdots, a_{k-1})v \\ &= b \text{ by the definition of } b \\ &= -a_kv + \gcd(b, a_k) \text{ by rearranging the EEA} \end{aligned}$$

Adding a_kv to both sides of this equation results in

$$a_1u_1 + a_2u_2 + \cdots + a_{k-1}u_{k-1} + a_kv = \gcd(b, a_k)$$

Since $b = \gcd(a_1, \cdots, a_{k-1})$,

$$\begin{aligned} \gcd(b, a_k) &= \gcd(\gcd(a_1, \cdots, a_{k-1}), a_k) \\ &= \gcd(a_1, a_2, \cdots, a_{k_1}, a_k) \end{aligned}$$

$\square$

Problem 3. Find all values of x between 0 and $m-1$ that are solutions of the following congruences. (*Hint*. If you can't figure out a clever way to find the solution(s), you can just substitute each value $x = 1, x = 2, \ldots, x = m-1$ and see which ones work.)

Subproblem 1. $x^2 \equiv 3 \mod 11$

Solution:
The squares modulo 11 are $\{0, 1, 4, 9, 5, 3, 3, 5, 9, 4, 1\}$
Therefore, $5^2 \equiv 3 \mod 11$ and $6^2 \equiv 3 \mod 11$

Subproblem 2. $x^2 \equiv 2 \mod 13$

Solution:
The squares modulo 13 are $\{0, 1, 4, 9, 3, 12, 10, 10, 12, 3, 9, 4, 1\}$
Therefore, $x^2 \equiv 2 \mod 13$ has no solutions

Problem 4. Suppose that $g^a \equiv 1 \mod m$ and that $g^b \equiv 1 \mod m$. Prove that

$$g^{\gcd(a,b)} \equiv 1 \mod m$$

Solution:
According to the Extended Euclidean Algorithm, $\exists u, v \in \mathbb{Z}$ such that $au + bv = \gcd(a, b)$. Then

$$g^{\gcd(a,b)} = g^{au+bv} = (g^a)^u \cdot (g^b)^v \equiv 1^u \cdot 1^v \equiv 1 \mod m$$

Since $g^a \equiv 1 \mod m$ and $g^b \equiv 1 \mod m$, the above equation holds.

Problem 5. Let $m \in \mathbb{Z}$

Subproblem 1. Suppose that m is odd. What integer between 1 and $m-1$ equals $2^{-1} \mod m$?

Solution:
Since m is odd, then $\dfrac{m+1}{2}$ must also be an integer, since $m+1$ is an even number. Therefore

$$2 \cdot \frac{m+1}{2} = m+1 \equiv 1 \mod m$$

Therefore $\dfrac{m+1}{2} = 2^{-1} \mod m$

Subproblem 2. More generally, suppose that $m \equiv 1 \mod b$. What integer between 1 and $m-1$ is equal to b^{-1} mod m?

Solution:
The assumption that $m = 1 \mod b \to b|(m-1)$ by proposition.
By the definition of division, $b|(m-1) \to \exists x \in \mathbb{Z}$ such that $m-1 = bx$. This requires that $\dfrac{m-1}{b} \in \mathbb{Z}$
Multiplying this fraction by b results in

$$b \cdot \frac{m-1}{b} = m-1 \equiv -1 \mod m$$

Multiplying this result by -1 results in

$$b \cdot \frac{1-m}{b} = 1-m \equiv 1 \mod m$$

However, $\dfrac{1-m}{b}$ is negative, but we can add multiples of m without effecting the value of modulo m. As a result, $\dfrac{1-m}{b} + m = \dfrac{1+(b-1)m}{b}$ is an integer and more importantly is congruent to $1 \mod m$.

Therefore, $\dfrac{1+(b-1)m}{b} = b^{-1} \mod m$

Problem 6. Consider the congruence

$$ax \equiv c \mod m$$

Prove that there is a solution if and only if $\gcd(a, m)|c$.

Solution:
First we prove that if $ax \equiv c \mod m$ has a solution, then $\gcd(a, m)|c$

Proof. Let $g = \gcd(a, m)$.
According to the definition of divisors, $\exists p, q \in \mathbb{Z}$ such that $a = gp$ and $m = gq$
We can take this definition and substitute it into the congruence which results in.

$$gpx \equiv c \mod gq$$

Since g divides both sides of the congruence, we can divide by g

$$px \equiv \frac{c}{g} \mod q$$

$\frac{c}{g}$ must be an integer in order for the congruence to hold.
This implies that $\gcd(a, m)|c$

This proves that if $ax \equiv c \mod m$ then $\gcd(a, m)|c$ □

Next we prove that if $\gcd(a, m)|c$, then $ax \equiv c \mod m$ has a solution

Proof. According to the Extended Euclidean Algorithm, $\exists u, v \in \mathbb{Z}$ such that

$$au + mv = \gcd(a, m)$$

$\gcd(a, m)|c \to (au + mv)|c$
By definition of divisibility, $\exists y \in \mathbb{Z}$ such that $c = (au + mv)y$
Rearranging this equation results in $mvy = c - auy \Rightarrow -mvy = auy - c$ or equivalently $mY = aX - c$ where $X = uy, Y = -vy$
Since $X, Y \in \mathbb{Z}$, the definition of divisibility implies that $m|(aX - c)$.
By one of the modular propositions, $m|(aX - c) \to aX \equiv c \mod m$.
This proves that if $\gcd(a, m)|c$, then $ax \equiv c \mod m$ has a solution □

Problem 7. Let

$a = 1234567890123456789012345678901234567890123456789012345678901234567890123456789,$
$b = 234567890123456789012345678901234567890123456789012345678901234567890123456789$

Find the inverse of a modulo b AND the inverse of b modulo a (as an integer between 0 and $b-1$ in the first case, and as an integer between 0 and $a-1$ in the second case), or explain why these inverses do not exist.

Solution:

```
def gcdExtended(a, b):
    if (a == 0):
        return b, 0, 1

    g, v, u = gcdExtended(b % a,a)
    return g, u - (b // a) * v, v

def modInverse(a, m):
    g, u, v = gcdExtended(a, m)
    if (g != 1):
        raise Exception("No Modular Inverse")
    return u % m

def main():
    a = 123456789012345678901234567890123456789012345678901234567890123456789012
    b = 234567890123456789012345678901234567890123456789012345678901234567890123

    aInv = modInverse(a, b)
    bInv = modInverse(b, a)

    print(f"a^(-1) mod b = {aInv}")
    print(f"a*a^(-1)==1 mod b? {(a * aInv) % b == 1}")

    print(f"b^(-1) mod a = {bInv}")
    print(f"b*b^(-1)==1 mod a? {(b * bInv) % a == 1}")

if __name__ == "__main__":
    main()
```

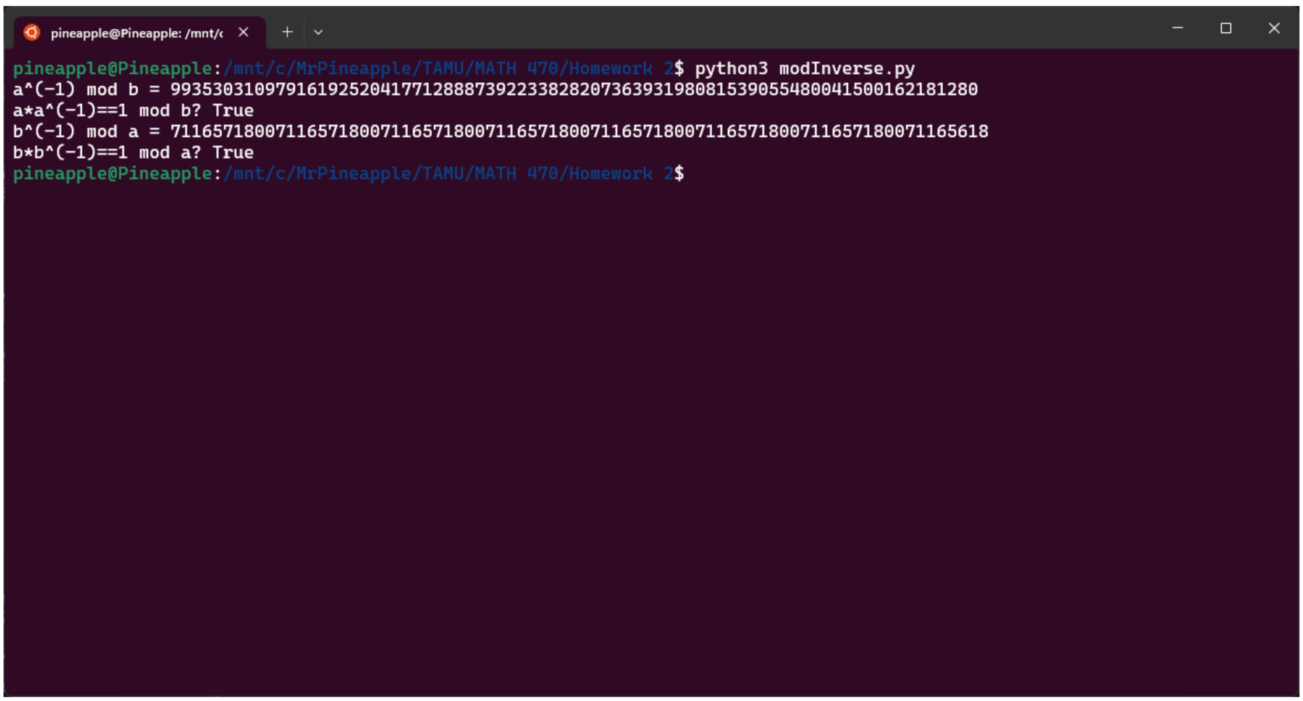

Figure 1: Output