Homework 3: GCD, Modular Inverses, RSA, and Recurrences

Instructions: **Due 01/31/25 11:59pm**

- Please type your solutions using LaTex or any other software. Handwritten solutions will not be accepted.
- Please try to write concise responses.
- You should not use pseudocode to describe your algorithms.
- Unless otherwise stated, saying log means base 2

Q1 The Euclidean algorithm for computing the greatest common divisor of two numbers is shown below.

Algorithm: Euclidean Algorithm for GCD

GCD(A, B)**:**

```
a := A
b := B
while b ≠ 0 do
    a, b := b, a mod b
return a
```

(a) Using the Euclidean algorithm, show that $GCD(44, 17) = 1$.

We use the Euclidean algorithm as follows,

$$\begin{aligned} 44 &= 17 \times 2 + 10 \\ 17 &= 10 \times 1 + 7 \\ 10 &= 7 \times 1 + 3 \\ 7 &= 3 \times 2 + 1 \\ 3 &= 1 \times 3 + 0 \end{aligned}$$

So we have run the algorithm till $b = 0$ at which the value of a is 1. Hence we see using the algorithm that $GCD(44, 17) = 1$

(b) Find whole numbers x and y such that $44x + 17y = 1$.

Hint: Use the results of the Euclidean algorithm from part (a). Solve for each remainder and substitute, working backwards until you can express 1 as a linear combination of 44 and 17.

Working backwards we have the following, Starting from the end we have,

$$7 = 3 \cdot 2 + 1 \implies 3 = \frac{7 - 1}{2}$$

So we can replace 3 in the previous one to get,

$$10 = 7 \cdot 1 + \frac{7 - 1}{2} \implies 10 \cdot 2 = 7 \cdot 3 - 17 = \frac{10 \cdot 2 + 1}{3}$$

Now we replace 7 in the previous one to get,

$$17 = 10 \cdot 1 + \frac{10 \cdot 2 + 1}{3} \implies 17 \cdot 3 = 10 \cdot 5 + 1 \implies 10 = \frac{17 \cdot 3 - 1}{5}$$

Now replace 10 in the previous one to get,

$$44 = 17 \cdot 2 + \frac{17 \cdot 3 - 1}{5} \implies 44 \cdot 5 = 17 \cdot 13 - 1$$

So rearranging we get,

$$44 \cdot -5 + 17 \cdot 13 = 1$$

Or that $x = -5$ and $y = 13$

(c) Using your solution from part (b), find z such that $z \cdot 17 \equiv 1 \mod 44$.

We need z such that,

$$z \cdot 17 \equiv 1 \mod 44$$

This means that we need k such that,

$$z \cdot 17 - 1 = k \cdot 44$$

Or we need to solve the equation,

$$(-k) \cdot 44 + z \cdot 17 = 1$$

We already know from the previous question that the solution is, if $-k = -5 \implies k = 5$ and $z = 13$

So we have when $z = 13$ that,

$$13 \cdot 17 \equiv 1 \mod 44$$

(d) Show that if a and N are integers such that $GCD(a, N) = 1$, then there always exists an integer x such that $ax \equiv 1 \mod N$ (i.e., a has a multiplicative inverse modulo N).

Hint: *Bézout's identity:* If $d = \gcd(a, b)$, then there exist integers x and y such that $ax + by = d$.

If $GCD(a, b) = d$ then bezouts identity tell us that there exists x, y such that,

$$ax + by = d$$

We are given that $gcd(a, N) = 1$ so we have x, y such that,

$$ax + Ny = 1$$

Rearranging the terms we have,

$$ax - 1 = Ny$$

Now by the definition of mod we have,

$$ax - 1 = Ny \implies ax - 1 \equiv 0 \mod N \implies ax \equiv 1 \mod N$$

Q2 The process used in Q1, part (b) is known as the *Extended Euclidean Algorithm*. It is extremely helpful for computing multiplicative inverses, as shown in parts (c) and (d) above. Below is the recursive implementation of the algorithm.

Algorithm: Extended Euclidean Algorithm

Extended-Euclid(a, b)**:**

Input: Two positive integers a and b with $a \geq b \geq 0$.
Output: Integers x, y, and d such that $d = \gcd(a, b)$ and $ax + by = d$.
if $b = 0$ **then**
  return $(1, 0, a)$
$(x', y', d) :=$ **Extended-Euclid**$(b, a \mod b)$
return $(y', x' - \lfloor \frac{a}{b} \rfloor y', d)$

(a) Write the recurrence relation for the algorithm in terms of $T(a, b)$. Analyze the time and space complexity in terms of n, where n is the number of bits in a.

At each function call we have to compute $a \mod b$ and we also recursively call the same function so our recurrence relation will look like,

$$T(a, b) = T(b, a \mod b) + c$$

this is assuming that computing mod would take a constant time, however if we take the number of bits into consideration using the naive approach computing $a \mod b$ would take $O(n^2)$ and as $n = \log(a)$ we have,

$$T(a, b) = T(b, a \mod b) + O(n^2) = T(b, a \mod b) + O(\log(a)^2)$$

Now to analyze the space and time complexity first we see how many time the recurrence actual will run for a given a, b. Now we have two cases to consider,

$$b \leq a/2 : a \mod b \leq a/2 \implies \text{we half } b \text{ in the worst case scenario}$$
$$b \geq a/2 : a \mod b = a - b \leq a/2 \implies \text{we half } b \text{ in the worst case scenario}$$

So in both cases we at least half b and we're iterating until $b = 0$. Hence the number of recurrences (the depth) of our algorithm would be in the worst case $\log(a)$.

So the time complexity would be $O(\log(a)^3) = O(n^3)$ as we are also computing the mod in each of those n function calls.

The space complexity would be $O(n)$ as in each call we only need to store a constant number of variables and we can forget the ones from the previous call. In addition in each function call all the operations we do mod by b so when we call we call by passing $a \mod b$ which ensures that it's always smaller than b. This ensures that it's always can be stored using just n bits. Hence it will never go above. So from the function beginning when we pass a, b all the stored numbers after that will always be smaller than a, b.

(b) Show that if a has a multiplicative inverse modulo N, then this inverse is unique (modulo N).

We are given that a has a multiplicative inverse modulo N. Now lets assume that the inverses were not unique mod N and are k_1, k_2 which means that,

$$a \cdot k_1 \equiv 1 \mod N \implies a \cdot k_1 - 1 = m_1 \cdot N \text{ for some } m_1 \in Z$$

and

$$a \cdot k_2 \equiv 1 \mod N \implies a \cdot k_2 - 1 = m_2 \cdot N \text{ for some } m_2 \in Z$$

So we have,

$$\begin{aligned} 1 = ak_1 - m_1N &= ak_2 - m_2N \\ ak_1 - ak_2 &= N(m_2 - m_1) \\ ak_1 - ak_2 &\equiv 0 \mod N \\ k_1 - k_2 &\equiv 0 \mod N \\ k_1 &\equiv k_2 \mod N \end{aligned}$$

So we have k_1, k_2 are equal mod N.

(c) How many integers modulo 11^3 have inverses? (Note: $11^3 = 1331$).

Hint: Euler's totient function, $\phi(n)$, counts the number of integers $1 \leq a \leq n$ that are coprime to n. It can be computed using the formula: $\phi(n) = n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)\cdots\left(1 - \frac{1}{p_k}\right)$, where each p_i is a distinct prime number dividing n.

If an integer n mod 11^3 has an inverse it means that the $GCD(11, n) = 1$ or that 11 and n are coprime. So it is enough to count the number of integers coprime to 11^3. We can use the Euler's totient functino for this to compute $\phi(1331)$. The distinct primes of $N = 1331$ are just 11. So we have,

$$\phi(1331) = 1331(1 - \frac{1}{11}) = 1331 \cdot \frac{10}{11} = 121 \cdot 10 = 1210$$

So we have a total of 1210 integers that have inverses modulo 11^3

Q3 RSA is a methodology to send and read secure messages via encryption and decryption. This is the paradigm. Alice wants to send a message x to Bob. Bob chooses prime numbers p and q, and an encryption key e. He computes $d = e^{-1} \mod (p-1)(q-1)$. Bob sends $N = pq$ and e to Alice. Alice computes $y = x^e \mod N$ and sends it to Bob. Bob computes $y^d \mod N$ to recover x.

(a) Prove that for a public key (N, e) and private key d for a message x, that $(x^e)^d \equiv x \mod N$

Hint: Try to first find $(x^e)^d \mod p$ and $(x^e)^d \mod q$ using Fermat's Little Theorem

First we have,

$$d \equiv e^{-1} \mod (p-1)(q-1)$$

This gives us,

$$ed \equiv 1 \mod (p-1)(q-1)$$

or $\exists k$ such that,

$$ed = k(p-1)(q-1) + 1$$

Now fermats little theorem tell us that,

$$x^{p-1} \equiv 1 \mod p, \text{ which is true because x and p are coprime}$$

So we have,

$$x^{ed} = x^{k(p-1)(q-1)+1} = xx^{k(p-1)(q-1)}$$

Now $x^{k(q-1)}$ is also coprime with p as p is a prime number so,

$$(x^{k(q-1)})^{p-1} \equiv 1 \mod p$$

But the left term is equivalent to $x^{ed} - x$

so we have,

$$x^{ed}x^{-1} \equiv 1 \mod p$$
$$x^{ed} \equiv x \mod p \implies p | x^{ed} - x$$

Similarly we can argue that,

$$x^{ed} \equiv x \mod q \implies q | x^{ed} - x$$

If $x^{ed} - x = a$ we now have $p|a$ and $q|a$ such that p and q are coprime. We know that for any divisors of a the LCM of those divisors will also divide a in our case as p and q are coprime we have $LCM(p, q) = pq = N$ which means that we have,

$$N|a \text{ or that } N|x^{ed} - x \text{ or that } x^{ed} - x \equiv 0 \mod N$$

So we have,

$$x^{ed} - x \equiv 0 \mod N \implies x^{ed} \equiv x \mod N$$

(b) Explain why e must be coprime with $(p-1)(q-1)$, where $N = pq$ and p, q are prime numbers in a public key (N, e), in order to be able to decipher the message using the private key. With this in mind, if $p, q > 2$, explain why e can never be even.

We need e to be coprime to $(p-1)(q-1)$ for us to be able to compute a d such that,

$$ed \equiv 1 \mod (p-1)(q-1)$$

If e wasn't coprime to it then it wouldn't have an inverse mod $(p-1)(q-1)$ and we need the unique d to be able to recover back to x.

Because p, q are both prime they are odd numbers and, $(p-1)(q-1)$ would end up being even. So if e was a factor for 2 then e and $(p-1)(q-1)$ would not be coprime and there wouldn't exist an inverse for e. Hence e cannot be a factor of 2.

(c) Let's say for example that $p = 31, q = 59,$ and $e = 7$. Find the value of d and show your work.

We know that,

$$d = e^{-1} \mod (p-1)(q-1)$$

Or that,

$$ed \equiv 1 \mod (p-1)(q-1)$$

We have $(p-1)(q-1) = 1740$

So we need to find the inverse of 7 mod 1740 or x such that,

$$x \cdot 7 \equiv 1 \mod 1740$$

$$7x - 1 = 1740k$$

The eucdlian algorithm gives us,

$$1740 \cdot 2 + 7 \cdot -497 = 1$$

So our solution is,

$$-497 \equiv 1243 \mod 1740$$

So our answer is 1243

Q4 *Euler's Theorem:* For any integers a and n that are relatively prime (coprime),

$$a^{\phi(n)} \equiv 1 \pmod{n}$$

where $\phi(n)$ is Euler's totient function.

Special Case — Fermat's Little Theorem: If p is prime and a is not divisible by p, then:

$$a^{p-1} \equiv 1 \pmod{p}$$

Use Euler's Theorem to find the remainder upon division of n by m, where:

(a) $n = 29^{202}$, $m = 13$

First we have $\phi(13) = 13 \cdot (1 - \frac{1}{13}) = 12$

So we know that,

$$a^{12} \equiv 1 \mod 13$$

We want to compute,

$$n \mod m = 29^{202} \mod 13 \text{ or } k \text{ such that,}$$

$$29^{202} \equiv k \mod 13$$

The closets power of 12 to 202 is 196 so we have,

$$29^{202} = 29^{192+10} = 29^{192}29^{10}$$

We know that 29^{16} is coprime with 13 so

$$(29^{16})^{12} \equiv 1 \mod 13$$

Or,

$$(29^{16})^{12}(29^{10}) \equiv (29^{10}) \mod 13$$

So our solution would be equivalent to,

$$29^{10} \mod 13$$

Because $29 \mod 13 = 3$ it is equivalent to,

$$3^{10} \mod 13$$

So see that for powers or three we have,

$$3 \equiv 3 \mod 13$$

$$3^2 \equiv 9 \mod 13$$

$$3^3 \equiv 1 \mod 13$$

$$3^4 \equiv 3 \mod 13$$

So it cycles, hence our solution would be when it goes back to the first which is equal to 3.

(b) $n = 99^{999999}$, $m = 23$

We have $\phi(23) = 22$ as 23 is a prime number.

And $999999 = 45454 \cdot 22 + 11$

So,

$$99^{999999} = 99^{45454 \cdot 22 + 11} = 99^{45454 \cdot 22} + 99^{11}$$

Now 99^{45454} is coprime with 23 because 99 is coprime with 23 and there are no other factors aside from that of 99 in that number so we have,

$$(99^{45454})^{22} \equiv 1 \mod 23$$

Multiplying 99^{11} on both sides we get,

$$99^9 99999 \equiv 99^{11} \mod 23$$

So our solution would be equivalent to,

$$99^{11} \mod 23$$

But we have, $99 \mod 23 = 7$ so we just need to compute,

$$7^{11} \mod 23$$

We see the following pattern,

$$7 \equiv 7 \mod 23$$

$$7^2 \equiv 3 \mod 23$$

$$7^4 \equiv 3^2 \equiv 9 \mod 23$$

$$7^8 \equiv 9^2 \equiv 12 \mod 23$$

$$7^8 7^3 \equiv 12 \times 21 \mod 23$$

And we have,

$$252 \mod 23 = 22$$

Which is our answer

(c) $n = 29^{198}$, $m = 20$

We have

$$\phi(20) = 20 \cdot (1 - 1/2)(1 - 1/5) = 8$$

And $198 = 8 \times 24 + 6$

We know that, 29^{24} is coprime with 20 because 29^{24} only has 29 as its distinct prime factor. So we have,

$$(29^{24})^8 \equiv 1 \mod 20$$

And similarly,

$$29^{198} \equiv 29^6 \mod 20$$

So our solution is,

$$29^6 \mod 20$$

We also have,

$$29 \equiv 9 \mod 20$$

So we can compute,

$$9^6 \mod 20$$

We have,

$$9 \equiv 9 \mod 20$$
$$9^2 \equiv 1 \mod 20$$
$$9^6 \equiv 1 \mod 20$$

So our solution is 1

(d) $n = 3^{1000000}$, $m = 14$

We have $\phi(14) = 14 \times (1 - 1/2)(1 - 1/7) = 6$

And $1000000 = 6 * 166666 + 4$

So we have,

$$3^{1000000} = 3^{6 \times 166666} + 3^4$$

And because 3^{166666} is coprime with 14 then, we know that,

$$3^{1666666} \equiv 1 \mod 14$$

So,

$$3^{1000000} \equiv 3^4 \mod 14$$

So our solution is,

$$3^4 \mod 14$$

We also have,

$$3 \equiv 3 \mod 14$$
$$3^2 \equiv 9 \mod 14$$
$$3^4 \equiv 11 \mod 14$$

So our solution is 11

Q5 Given two points $x, y \in \mathbb{R}^2$, we say that the pair x and y are "close" if their distance $\|x - y\|_2 = 1$. We want to construct n (distinct) points such that the number of "close" pairs of points is $\Omega(n \log n)$. For simplicity, you may assume n is a power of 2.

(a) Describe an efficient algorithm to construct such a set of points.

Hint: Here is a visualizer tool that shows a possible algorithm. If you are going to use a similar strategy, make sure your algorithm does not create any overlaps. `https://nickpapciak.github.io/close-pairs-visualizer/`.

We create a recursive algorithm and start with the case of n. The algorithm calls itself to generate points for the case for $n/2$ and using the results of that it generates $n/2$ more points for each of the generated $n/2$ points. We also keep track of the unit vector angles that we used to distance each point (this angle will be relative to the origin. So for instance if after the first point at the origin we generated the next point at $(\sqrt{2}, \sqrt{2})$ then we store $\pi/4$) we also make sure this list is sorted (Essentially we're storing all the angles we've placed points so that the new points we place will not be in the same angle. We can store these angles by adding to a min or max heap.). Now for each of the $n/2$ points we create another point that is of distance of unit 1. To determine where to place these points we'll take the two smallest elements from our heap which will take $\log n$ time and then we'll find the angle between those two. As these two are the smallest angles the angle bnetwen them will be unique as if it was already in the heap then it would be the 2nd smallest and would be chosen. Now relative to each $n/2$ points we place another point at this angle relative to the positive x-axis. This ensures that there will be no overlap.

(b) Show why your algorithm is correct.

Hint: Write a recurrence for the number of close pairs and solve for it *exactly*.

Our recurrence would be,

$$C(n) = 2C(n/2) + \frac{n}{2}$$

As we're placing the new points in the exact same relative angles we are doubling the current number of unit distances and adding extra $n/2$ distances.

We have,

$$\begin{aligned} C(n) &= 2(2(C(n/4) + n/4) + n/2 \\ &= 2^2 C(n/4) + n/2 + n/2 \end{aligned}$$

As we are halving n the depth is $\log_2(n)$ so as we unroll our recurrence we have

$$\begin{aligned} C(n) &= n/2 + n/2 + \ldots \ \log n \text{ times} \\ &= \frac{n}{2} \log n \end{aligned}$$

which is the number of close points.

(c) Analyze the time complexity of your algorithm.

Our recurrence for the algorithm is,

$$T(n) = T(n/2) + O(n)$$

Here we are subdividing our problem by 2 to get $T(n/2)$ and we also have n work to create the n new points. To find the 2 smallest and add to our heap we just need $O(\log n)$ which would be $O(n)$. So using the master theorem we have, $d = 1$ and $a = 1, b = 2$. We see that $log_2 1 = 0 < 1$ or that $log_b a < d$ this means that our complexity is,

$$T(n) = O(n)$$