

Homework 1 Solutions - Sri Raga Velagapudi
Algebra Section

1. Show that if $n \in \mathbb{Z}$ then for every integer a with $\gcd(a, n) = 1$, there exists a unique $x \pmod n$ such that $ax = 1 \pmod n$.

By the definition of gcd, for a given a, n and $d, \gcd(a, n) = d = xa + yn$ where $x, y \in \mathbb{Z}$. In our case, $xa + yn = 1 = \gcd(a, n)$. $(xa + yn) \pmod n = 1 \pmod n \Rightarrow xa \pmod n = 1 \pmod n \Rightarrow$ In the ring \mathbb{Z}_n $xa = 1 \Rightarrow x = a^{-1}$. By the definition of the ring \mathbb{Z}_n , x is unique.

2. Here is an age-guessing game you might play at a party. You ask a fellow party-goer to divide his age by each of the numbers a, 4 and 5 and tell you the remainders. Show how to use this information to determine the age.

Let the party goer's age be x . Then we have $x = r_3 \pmod 3, x = r_4 \pmod 4, x = r_5 \pmod 5$. This means $x = (r_3m_3 + r_4m_4 + r_5m_5) \pmod 60$ where:

$$m_3 = s_3 \times 20 \text{ where } s_3 = 20^{-1} \pmod 3 = 2$$

$$m_4 = s_4 \times 15 \text{ where } s_4 = 15^{-1} \pmod 4 = 3$$

$$m_5 = s_5 \times 12 \text{ where } s_5 = 12^{-1} \pmod 5 = 3$$

$$\Rightarrow m_3 = 40, m_4 = 45, m_5 = 36$$

$$\Rightarrow x = (40r_3 + 45r_4 + 36r_5) \pmod 60$$

3. For any integer $n > 1$, let $\phi(n)$ denote the number of positive integers less than n and coprime to n . If m, n are coprime then $\phi(m, n) = \phi(m) \cdot \phi(n)$. If n is a prime power, $n = p^e$ where $\phi(n) = (p^e - p^{e-1})$. Use the two results above to deduce a formula for $\phi(n)$ in terms of the prime factorization of n .

Let p_1, \dots, p_k be the set of unique prime factors of n , then we have $n = p_1^{e_1} \cdot p_2^{e_2} \dots p_k^{e_k}$ where e_i us the appropriate power for the prime factor p_i .

$$\phi(n) = \phi(p_1^{e_1}) \dots \phi(p_k^{e_k}) = \prod_{i=1}^{i=k} \phi(p_i^{e_i}) = \prod_{i=1}^{i=k} [p_i^{e_i} - p_i^{e_i-1}].$$

4. For each pair of rings below, determine if they are isomorphic. If they are not isomorphic prove it, else provide an isomorphism.

$$4-1: R_1 = \mathbb{Z}_{15}, R_2 = \mathbb{Z}_3 \oplus \mathbb{Z}_5$$

$R_1 \cong R_2$. Let ϕ be the isomorphism. For $a \in \mathbb{Z}_{15}$, $\phi(a) = (a \pmod 3, a \pmod 5)$. For $\phi^{-1}(\alpha, \beta) = \alpha(5(5^{-1} \pmod 3)) + \beta(3(3^{-1} \pmod 5)) = 10\alpha + 6\beta$.

The above definition of the isomorphism is bijective since $\phi^{-1}(\alpha, \beta) \pmod 3 = \alpha$ and $\phi^{-1}(\alpha, \beta) \pmod 5 = \beta$.

By the properties of modulo we have $\phi(ab) = (ab \pmod 3, ab \pmod 5) = (a \pmod 3, a \pmod 5) * (b$

$\text{mod } 3, b \text{ mod } 5) = \phi(a)\phi(b)$.

Similarly $\phi(a + b) = (a + b \text{ mod } 3, a + b \text{ mod } 5) = (a \text{ mod } 3, a \text{ mod } 5) + (b \text{ mod } 3, b \text{ mod } 5) = \phi(a) + \phi(b)$.

4-2: $R_1 = \mathbb{Z}_{25}, R_2 = \mathbb{Z}_5 \oplus \mathbb{Z}_5$

$R_1 \not\cong R_2$. If R_1 and R_2 are isomorphic, then corresponding elements must have the same order. Note that in $\mathbb{Z}_5 - \{0\} = \{1, 2, 3, 4\}$ each element x has $x^4 = 1$. Then if we look at the set $\mathbb{Z}_5 \oplus \mathbb{Z}_5$ without any elements containing 0, then we have that every $(a, b)^4 = (1, 1)$. However, this does not hold true for \mathbb{Z}_{25} . $2 \text{ mod } 25$ in \mathbb{Z}_{25} , for example, does not match this expectation. $(2 \text{ mod } 25)^4 = 16 \text{ mod } 25$ which does not equal $1 \text{ mod } 25$.

4-3: $R_1 = \mathbb{Q}[x]/\langle x^2 - 7 \rangle, R_2 = \mathbb{Q}[y]/\langle y^2 - 2 \rangle$

$R_1 \not\cong R_2$. Both $x^2 - 7$ and $y^2 - 2$ are irreducible over $\mathbb{Q}[x]$. Note that all elements in R_1 with $x = 0$ must map to those in R_2 with $y = 0$.

In R_1 , $x^2 - 7 = 0$, this implies that there is an element, $g \in R_1$ such that $g^2 = 7$. In order for $R_1 \cong R_2$, one such element must exist in R_2 as well. Let that element be $(a + by)$, then we have $7 = (a + by)^2 = a^2 + b^2y^2 + 2aby = (a^2 + 2b^2) + (2ab) \cdot y$. This means that $a^2 + 2b^2 = 7$ and that $2ab = 0$. Solving, we have $b = \sqrt{7}/2$. Since the constants in R_2 are all in \mathbb{Q} , b cannot exist. Therefore no element $z \in R_2$ exists such that $z^2 = 7$. Therefore $R_1 \not\cong R_2$.

4-4: $R_1 = \mathbb{R}[x]/\langle x^2 - 1 \rangle, R_2 = \mathbb{R}[x]/\langle x^2 + 1 \rangle$

$R_1 \not\cong R_2$. $x^2 - 1$ is reducible over $\mathbb{R}[x]$ while $x^2 + 1$ is not reducible over $\mathbb{R}[x]$. By CRT, $R_1 \cong \mathbb{R}[x]/\langle x - 1 \rangle \oplus \mathbb{R}[x]/\langle x + 1 \rangle$. However, R_2 cannot be isomorphic to a similar direct sum of co-primes since $x^2 + 1$ is irreducible in $\mathbb{R}[x]$. This is a fundamental property difference. Therefore $R_1 \not\cong R_2$.

4-5: $R_1 = \mathbb{C}[x]/\langle x^2 - 1 \rangle, R_2 = \mathbb{C}[x]/\langle x^2 + 1 \rangle$

$R_1 \cong R_2$. All constants in R_1 must map to constants in R_2 . Let's map $\varphi : x \mapsto ix$. Then we have that following:

$$\varphi(a + bx) = a + ibx$$

$$\begin{aligned} \varphi((a+bx)(c+dx)) &= \varphi(ac+bd+(ad+bc)x) = (ac+bd) + i(ad+bc)x = ac + ix(ad+bc) - bdx^2 \\ &= ac + iadx + ibcx - bdx^2 = (a + ibx)(c + idx) = \varphi(a + bx)\varphi(c + dx) \end{aligned}$$

$$\begin{aligned} \text{Similarly, } \varphi((a+bx)+(c+dx)) &= \varphi((a+c)+(b+d)x) = (a+c) + i(b+d)x = (a+ibx) + (c+idx) \\ &= \varphi(a + bx) + \varphi(c + dx). \end{aligned}$$

With all the constants mapping to constants and $\varphi : x \mapsto ix$, bijectivity is satisfied. Therefore $R_1 \cong R_2$.

4-6: $R_1 = \mathbb{Q}[x]/\langle x^3 - 1 \rangle, R_2 = \mathbb{Q}[x]/\langle x^3 + x^2 - 2x \rangle$

$R_1 \not\cong R_2$. Over $\mathbb{Q}[x]$, $x^3 - 1$ has two irreducible factors, $x - 1$ and $x^2 + x + 1$. On the other hand, $x^3 + x^2 - 2x$ has three irreducible factors $x, x + 2$ and $x - 1$. By the CRT, this implies that $x^3 - 1$ would be isomorphic to a direct sum of two co-prime rings while $x^3 + x^2 - 2x$ would be isomorphic to a direct sum of three co-prime rings.

Since the co-rime ring $\mathbb{R}[x]/\langle x-1 \rangle$ is common for both sets of direct sums, the problem reduces to showing that $\mathbb{R}[x]/\langle x^2+x+1 \rangle$ is isomorphic to $\mathbb{R}[x]/\langle x \rangle \oplus \mathbb{R}[x]/\langle x-2 \rangle$. This is, however, not an isomorphism, since the direct sum would only contain constants. Therefore $R_1 \not\cong R_2$.

$$4-7: R_1 = \mathbb{Z}_7/\langle x^3 - 1 \rangle, R_2 = \mathbb{Z}_7[x]/\langle x^3 + x^2 - 2x \rangle$$

$R_1 \cong R_2$. In $\mathbb{Z}_7[x]$ we have three elements that satisfy the equation $x^3 - 1$. They are 1, 2 and 4. $x^3 + x^2 - 2x$ can be factored into $x(x-1)(x-5)$.

By CRT we have $\mathbb{Z}_7[x]/\langle x^3 - 1 \rangle \cong \mathbb{Z}_7[x]/\langle x-1 \rangle \oplus \mathbb{Z}_7[x]/\langle x-2 \rangle \oplus \mathbb{Z}_7[x]/\langle x-4 \rangle$ and $\mathbb{Z}_7[x]/\langle x^3 + x^2 - 2x \rangle \cong \mathbb{Z}_7[x]/\langle x \rangle \oplus \mathbb{Z}_7[x]/\langle x-1 \rangle \oplus \mathbb{Z}_7[x]/\langle x-5 \rangle$.

Since all polynomial divisors are of order 1, the direct sums for both rings are simply linear transformations. Therefore $R_1 \cong R_2$.

Complexity Section

1. Show that if a function $f(x)$ can be computed in polynomial time using an oracle for another polynomial time computable function $g(x)$ then $f(x)$ can be computed in polynomial time without the use of oracles.

For a given string β and a language B , oracles can tell if $\beta \in B$.

If $f(x)$ is using an oracle for $g(x)$ and is still a polynomial time function, then we must call any oracle at most polynomial number of times.

If all the calls within $f(x)$ using any result from the oracle were replaced directly with $g(x)$ and the resulting changed function evaluated at the necessary value, then the time taken would still be polynomial since the composition of two polynomial functions is still polynomial.

2. Show that if a function $f(x)$ has an NC algorithm using an oracle for another NC-computable function $g(x)$ then $f(x)$ itself can be computed in NC without using oracles.

Similar to problem 1, if all places in $f(x)$ using results from the oracle are replaced with $g(x)$ directly and then $f(x)$ would still have a polynomial size circuit because the sum of two polynomials is still a polynomial. Thus $f(x)$ would still be NC even if it did not use oracles. Note that the depth, however, would increase by $k(|g(x)|)$ where k is the number of places within $f(x)$ where $g(x)$ occurs and $|g(x)|$ is the size of the circuit for $g(x)$.

3. Solve the following recurrence relations in terms of n , expressing the resulting functions asymptotically:

3-1: $T(n) = 2.T(n/2) + cn$

Answer to 3-1

3-2: $T(n) = 3.T(n/5) + cn$

Answer to 3-2

3-3: $T(n) = 2.T(n - 1)$

$T(n) = 2.T(n - 1) = 2.2.T(n - 2) = 2.2.2.T(n - 3) = 2^4.T(n - 4) = \dots$. This continues until $n - x = 0 \Rightarrow x = n$. $T(n) = 2^n \Rightarrow T(n) = O(2^n)$.

Algorithmic Algebra Section

1. Let $a, b \in \mathbb{Z}$. Recall that if $\gcd(a, b) = d$, then there exists integers x, y s.t. $ax + by = d$. Devise an algorithm that given an a and b with $a > b$, computes d and an x and a y in time $O(|\langle a \rangle| |\langle b \rangle| + |\langle b \rangle|^3)$.

Given m_0 and m_1 , the euclidean algorithm for the $\gcd(m_0, m_1)$ proceeds by finding the remainder sequence $m_0, m_1, m_2, \dots, 0$ where $m_i = m_{i-2} \bmod m_{i-1}$ for $i > 1$. Then $\gcd(m_0, m_1)$ is the last non-zero m_i .

The extended euclidean algorithm works in the following manner:

Let q_i be the quotient of the i^{th} remaindering step. Then $m_{i+1} = m_{i-1} - q_i m_i$ ($i = 2, \dots, k-1$).

Let (s_0, \dots, s_k) and (t_0, \dots, t_k) be two sequences such that $m_i = s_i m_0 + t_i m_1$ ($i = 0, \dots, k$).

When $i = k$ we have $m_k = \gcd(m_0, m_1) = s_k m_0 + t_k m_1$.

Let $s_{i+1} = s_{i-1} - q_i s_i$ and $t_{i+1} = t_{i-1} - q_i t_i$ ($i = 2, \dots, k-1$).

Since $m_0 = 1m_0 + 0m_1$ and $m_1 = 0m_0 + 1m_1$, we have $s_0 = 1, t_0 = 0, s_1 = 0, t_1 = 1$ and $s_k = 0, t_k = 1$.

Inductively, $m_{i+1} = m_{i-1} - q_i m_i = (s_{i-1} m_0 + t_{i-1} m_1) - q_i (s_i m_0 + t_i m_1)$.

Time Taken:

To calculate the remainders and quotients for m_0, m_1 with $m_0 = a$ and $m_1 = b$ we take $|\langle a \rangle| |\langle b \rangle| + |\langle b \rangle| |\langle b \rangle|^2$ to do $a \bmod b$ and $b \bmod m_2, m_2 \bmod m_3$ so on $|\langle b \rangle|$ number of times. This is because with each consecutive mod we lose at least 1 bit.

The above calculations also generate all the q_i s necessary to calculate all s_i and t_i values.

There are at most $|\langle b \rangle|$ number of s_i and t_i values each. Each s_i calculation takes $|\langle b \rangle|^2 + |\langle b \rangle|$ for the one multiplication and one subtraction.

Similarly each t_i calculation takes $|\langle b \rangle|^2 + |\langle b \rangle|$ time.

Together, all s_i and t_i calculations take $|\langle b \rangle| (2|\langle b \rangle|^2 + 2|\langle b \rangle|)$. The total operation count: $|\langle a \rangle| |\langle b \rangle| + |\langle b \rangle|^3 + 2|\langle b \rangle|^3 + 2|\langle b \rangle|^2$.

1-1: Show that an analogous algorithm works for polynomials and assuming that all the underlying field operations are done in constant time, the time complexity for this is $O(\deg(a) * \deg(b))$.

$\deg(b) + \deg(b)^3$.

The extended Euclidean Algorithm would work in the following manner for polynomials:

Like for integers $q_1(x)$ would be the quotient of the i^{th} remaindering step. Then $m_{i+1}(x) = m_{i-1}(x) - q_i(x)m_i(x)$ for $i \in \{1, \dots, k-1\}$.

Similarly $(s_0(x), \dots, s_k(x))$ and $(t_0(x), \dots, t_k(x))$ would be the two polynomial sequence such that $m_i(x) = s_i(x)m_0(x) + t_i(x)m_1(x)$ for $i \in \{0, \dots, k\}$.

When $i = k$ we have $m_k(x) = \gcd(m_0(x), m_1(x)) = s_k(x)m_0(x) + t_k(x)m_1(x)$.

Let $s_{i+1}(x) = s_{i-1}(x) - q_i(x)s_i(x)$ and $t_{i+1}(x) = t_{i-1}(x) - q_i(x)t_i(x)$ for $i \in \{2, \dots, k-1\}$.

Since $m_0(x) = m_0(x) + 0m_1(x)$ and $m_1(x) = 0m_0(x) + m_1(x)$ we have $s_0(x) = 1, t_0(x) = 0, t_1(x) = 1, s_1(x) = 0$.

Inductively, similar to the proof given for integers we have $m_{i+1}(x) = s_{i+1}m_0 + t_{i+1}m_1$.

Assuming field operations are done in constant time, the following are the time complexity details.

The first mod: $a \bmod b$ takes time $\deg(a) * \deg(b)$.

Each subsequent mod takes less than $\deg(b) * \deg(b)$.

Maximum number of subsequent mods is $\deg(b)$. This is because each mod will result in a remainder of degree atmost one less than the divisor which could be atmost b .

As a result the total is $O(\deg(a) * \deg(b) + \deg(b)^3)$.

2. Show that given integers a, t, m we can compute $a^t \bmod m$ in time $O(|\langle a \rangle| * |\langle m \rangle| + |\langle m \rangle|^2 * |\langle t \rangle|)$.

$a^t \bmod m = (a \bmod m)^t \bmod m$.

The operatio $a \bmod m = r_0$ takes $|\langle a \rangle| * |\langle m \rangle|$ time. We now need to calculate $r_0^t \bmod m$.

Note that $r_0 < m$. Without loss of generality, we can assume that $r_0^2 > m$. Then we have: $r_0^t \bmod m = (r_0^2 \bmod m)^{t/2} \bmod m$.

Calculating $r_0^2 \bmod m = r_1$ takes $|\langle m \rangle|^2$ operations. Now we have $t/2$ number of r_1 remainders. We can repeat the same process and calculate $(r_1^2 \bmod m)^{t/4} \bmod m$. $r_1^2 \bmod m$ would once again take $|\langle m \rangle|^2$ operations.

Continueing this process we would have $t/2 + t/4 + t/8 + \dots$ number of $|\langle m \rangle|^2$ sized operations for calculating $r_i^2 \bmod m$. $t > t/2 + t/4 + t/8 + \dots$. Therefore we have that the total for calculating $r^t \bmod m = |\langle m \rangle|^2 * |\langle t \rangle|$.

The total time taken for calculating $a^t \bmod m$ is therefore $|\langle a \rangle| * |\langle m \rangle| + |\langle m \rangle|^2 * |\langle t \rangle|$.