Lecture 6: Chinese Remainder Theorem & Algorithm

Rafael Oliveira

University of Waterloo
Cheriton School of Computer Science

rafael.oliveira.teaching@gmail.com

January 27, 2021

Overview

- Background on Rings and Quotients
- Chinese Remainder Theorem
- Variants on Chinese Remaindering
- Conclusion
- Acknowledgements

Ring Basics

- Given a ring R, an *ideal* $I \subset R$ is a subset of the ring R such that:
  1. I is closed under addition

     $$a, b \in I \Rightarrow a + b \in I$$

  2. I is closed under multiplication by elements of R

     $$a \in I, s \in R \Rightarrow s \cdot a \in I$$

Ring Basics

- Given a ring R, an *ideal* $I \subset R$ is a subset of the ring R such that:
  1. I is closed under addition
  $$a, b \in I \Rightarrow a + b \in I$$
  2. I is closed under multiplication by elements of R
  $$a \in I, s \in R \Rightarrow s \cdot a \in I$$
- Examples:
  1. (0) is ideal generated by the 0 element of the ring

$0 + 0 = 0$

$r \in R \quad r \cdot 0 = 0$

Ring Basics

- Given a ring R, an *ideal* $I \subset R$ is a subset of the ring R such that:
  1. I is closed under addition
     $$a, b \in I \Rightarrow a + b \in I$$
  2. I is closed under multiplication by elements of R
     $$a \in I, s \in R \Rightarrow s \cdot a \in I$$
- Examples:
  1. (0) is ideal generated by the 0 element of the ring
  2. R is an ideal generated by $(1) = R$

$(g_1, \ldots, g_m) :=$ ideal generated by elements $g_1, \ldots, g_m$

Ring Basics

- Given a ring R, an *ideal* $I \subset R$ is a subset of the ring R such that:
  1. I is closed under addition

$$a, b \in I \Rightarrow a + b \in I$$

  2. I is closed under multiplication by elements of R

$$a \in I, s \in R \Rightarrow s \cdot a \in I$$

- Examples:
  1. (0) is ideal generated by the 0 element of the ring
  2. R is an ideal
  3. ring of integers $\mathbb{Z}$ then the set of all even numbers is the ideal generated by 2, denoted (2)

$$2k \ , \ k \in \mathbb{Z} \Rightarrow 2k \in (2)$$

Ring Basics

- Given a ring R, an *ideal* $I \subset R$ is a subset of the ring R such that:
  1. I is closed under addition
  $$a, b \in I \Rightarrow a + b \in I$$
  2. I is closed under multiplication by elements of R
  $$a \in I, s \in R \Rightarrow s \cdot a \in I$$
- Examples:
  1. (0) is ideal generated by the 0 element of the ring
  2. R is an ideal
  3. ring of integers $\mathbb{Z}$ then the set of all even numbers is the ideal generated by 2, denoted (2)
  4. In $\mathbb{Q}[x]$ the set of all polynomials whose constant coefficient is zero is the ideal (x) generated by x

$p(0) = p_0 = 0$

$$\underbrace{\{p(x) \in \mathbb{Q}[x] \mid p(0) = 0\}}_{\text{evaluation at point}} = \underbrace{(x)}_{\text{generators}}$$

$p(x) = p_1 x + \cdots + p_d x^d$

$x \mid p(x)$

Ring Basics

- Given a ring R, an *ideal* $I \subset R$ is a subset of the ring R such that:
  1. I is closed under addition
  $$a, b \in I \Rightarrow a + b \in I$$
  2. I is closed under multiplication by elements of R
  $$a \in I, s \in R \Rightarrow s \cdot a \in I$$
- Examples:
  1. (0) is ideal generated by the 0 element of the ring
  2. R is an ideal
  3. ring of integers $\mathbb{Z}$ then the set of all even numbers is the ideal generated by 2, denoted (2)
  4. In $\mathbb{Q}[x]$ the set of all polynomials whose constant coefficient is zero is the ideal (x) generated by x
  5. In $\mathbb{Q}[x, y]$ the set of all polynomials whose constant coefficient is zero is the ideal (x, y) generated by x and y

Quotient Rings

- Given a ring R, and an ideal $I \subset R$, we can form equivalence classes of elements of R modulo I

$$a \sim b \Leftrightarrow a - b \in I$$

$\mathbb{Z}_2$ integers modulo (2)

$3, 5$ $\qquad 3 \sim 5$

$5 - 3 = 2 \in (2)$

odd ~ 1 $\qquad \mathbb{Z}_2 = (\{0,1\}, +, \cdot)$

even ~ 0

Quotient Rings

- Given a ring R, and an ideal $I \subset R$, we can form equivalence classes of elements of R modulo I

$$a \sim b \Leftrightarrow a - b \in I$$

- If we only consider these equivalence classes, we have the *quotient ring* R/I

$$\mathbb{Z}_2 := \mathbb{Z}/\underbrace{2\mathbb{Z}}_{(2)}$$

Quotient Rings

- Given a ring R, and an ideal $I \subset R$, we can form equivalence classes of elements of R modulo I

$$a \sim b \Leftrightarrow a - b \in I$$

- If we only consider these equivalence classes, we have the *quotient ring* R/I
- Examples:
  1. $R = \mathbb{Z}$ and $I = (2)$ gives the field $\mathbb{Z}_2$

Quotient Rings

- Given a ring R, and an ideal $I \subset R$, we can form equivalence classes of elements of R modulo I

$$a \sim b \Leftrightarrow a - b \in I$$

- If we only consider these equivalence classes, we have the *quotient ring* R/I
- Examples:
  1. $R = \mathbb{Z}$ and $I = (2)$ gives the field $\mathbb{Z}_2$
  2. $R = \mathbb{Z}$ and $I = (6)$ gives the set of integers modulo 6, $\mathbb{Z}_6$

$\mathbb{Z}_6$ not field because

2 and 3 are zero divisors

$2 \cdot 3 = 0 \Rightarrow 2, 3$ do not

have inverse in $\mathbb{Z}_6 \Rightarrow$ not field

Quotient Rings

- Given a ring R, and an ideal $I \subset R$, we can form equivalence classes of elements of R modulo I

$$a \sim b \Leftrightarrow a - b \in I$$

- If we only consider these equivalence classes, we have the *quotient ring* R/I
- Examples:
  1. $R = \mathbb{Z}$ and $I = (2)$ gives the field $\mathbb{Z}_2$
  2. $R = \mathbb{Z}$ and $I = (6)$ gives the set of integers modulo 6, $\mathbb{Z}_6$
- An element $q \in R$ is *irreducible* if q is not a unit and $q = a \cdot b \Rightarrow$ either a or b are a unit.

divisor 1

2 irreducible

6 reducible $\quad 6 = 2 \cdot 3$

Quotient Rings

- Given a ring R, and an ideal $I \subset R$, we can form equivalence classes of elements of R modulo I

$$a \sim b \Leftrightarrow a - b \in I$$

- If we only consider these equivalence classes, we have the *quotient ring* R/I
- Examples:
  1. $R = \mathbb{Z}$ and $I = (2)$ gives the field $\mathbb{Z}_2$
  2. $R = \mathbb{Z}$ and $I = (6)$ gives the set of integers modulo 6, $\mathbb{Z}_6$
- An element $q \in R$ is *irreducible* if q is not a unit and $q = a \cdot b \Rightarrow$ either a or b are a unit.
- An ideal $I \subset R$ is *prime* if for any $\underline{a, b \in R}$, if $\underline{ab \in I}$ then $\underline{a \in I}$ or $\underline{b \in I}$

over $\mathbb{Z}$ prime and irreducible coincide

Quotient Rings

- Given a ring R, and an ideal $I \subset R$, we can form equivalence classes of elements of R modulo I

$$a \sim b \Leftrightarrow a - b \in I$$

- If we only consider these equivalence classes, we have the *quotient ring* R/I
- Examples:
  1. $R = \mathbb{Z}$ and $I = (2)$ gives the field $\mathbb{Z}_2$
  2. $R = \mathbb{Z}$ and $I = (6)$ gives the set of integers modulo 6, $\mathbb{Z}_6$
- An element $q \in R$ is *irreducible* if q is not a unit and $q = a \cdot b \Rightarrow$ either a or b are a unit.
- An ideal $I \subset R$ is *prime* if for any $a, b \in R$, if $ab \in I$ then $a \in I$ or $b \in I$

whole ring

- Two ideals $I, J \subset R$ are *coprime* if $I + J = R$

also an ideal

a, b coprime $\iff \gcd(a,b) = 1$

over $\mathbb{Z}$ Extended Euclidean Algorithm $\gcd(a,b) = sa + tb$

$I = (a) \quad J = (b)$

$I + J \ni \gcd(a,b)$

$sa \quad tb$

$1 \in I + J \Rightarrow I + J = R$.

Unique Factorization Domains

domain: ring R with no zero divisor.

- An integral domain R is a *unique factorization domain* (UFD) if
  1. every element in R is expressed as a product of finitely many irreducible elements
  2. Every irreducible element $p \in R$ yields a prime ideal (p)

Unique Factorization Domains

- An integral domain R is a *unique factorization domain* (UFD) if
  1. every element in R is expressed as a product of finitely many irreducible elements
  2. Every irreducible element $p \in R$ yields a prime ideal (p)
- A very special kind of UFD, which we have seen a lot, is a *principal ideal domain* (PID): R is a PID if every ideal of R is principal (generated by *one element*)

Unique Factorization Domains

- An integral domain R is a *unique factorization domain* (UFD) if
  1. every element in R is expressed as a product of finitely many irreducible elements
  2. Every irreducible element $p \in R$ yields a prime ideal (p)
- A very special kind of UFD, which we have seen a lot, is a *principal ideal domain* (PID): R is a PID if every ideal of R is principal (generated by *one element*)
- Examples of PIDs and UFDs
  1. $\mathbb{Z}$ is a PID (and hence UFD)

Unique Factorization Domains

- An integral domain R is a *unique factorization domain* (UFD) if
  1. every element in R is expressed as a product of finitely many irreducible elements
  2. Every irreducible element $p \in R$ yields a prime ideal (p)
- A very special kind of UFD, which we have seen a lot, is a *principal ideal domain* (PID): R is a PID if every ideal of R is principal (generated by *one element*)
- Examples of PIDs and UFDs
  1. $\mathbb{Z}$ is a PID (and hence UFD)
  2. $\mathbb{Q}[x]$ is a PID (and hence UFD)

Unique Factorization Domains

- An integral domain R is a *unique factorization domain* (UFD) if
  1. every element in R is expressed as a product of finitely many irreducible elements
  2. Every irreducible element $p \in R$ yields a prime ideal (p)
- A very special kind of UFD, which we have seen a lot, is a *principal ideal domain* (PID): R is a PID if every ideal of R is principal (generated by *one element*)
- Examples of PIDs and UFDs
  1. $\mathbb{Z}$ is a PID (and hence UFD)
  2. $\mathbb{Q}[x]$ is a PID (and hence UFD)
  3. any Euclidean domain is a PID (and hence UFD)

Unique Factorization Domains

- An integral domain R is a *unique factorization domain* (UFD) if
  1. every element in R is expressed as a product of finitely many irreducible elements
  2. Every irreducible element $p \in R$ yields a prime ideal (p)
- A very special kind of UFD, which we have seen a lot, is a *principal ideal domain* (PID): R is a PID if every ideal of R is principal (generated by *one element*)
- Examples of PIDs and UFDs
  1. $\mathbb{Z}$ is a PID (and hence UFD)
  2. $\mathbb{Q}[x]$ is a PID (and hence UFD)
  3. any Euclidean domain is a PID (and hence UFD)
  4. $\mathbb{Q}[x, y]$ is a UFD but *not* a PID (x, y)

Gauss' lemma: R is UFD $\Leftarrow$ R[x] is UFD

Ring Homomorphisms

- A *homomorphism* between rings R, S is a map $\phi : R \to S$ *preserving the ring structure*
  1. $\phi(1) = 1$
  2. $\phi(a+b) = \phi(a) + \phi(b)$
  3. $\phi(ab) = \phi(a) \cdot \phi(b)$

$$\phi(1_R) = 1_S$$

$$\begin{tikzcd}
R \arrow[r, "\text{add}", "\text{mult.}"'] \arrow[d, "\phi"'] & R \arrow[d, "\phi"] \\
S \arrow[r, "\text{add}", "\text{mult.}"'] & S
\end{tikzcd}$$

diagram commutes

Ring Homomorphisms

- A *homomorphism* between rings R, S is a map $\phi : R \to S$ *preserving the ring structure*
  1. $\phi(1) = 1$
  2. $\phi(a + b) = \phi(a) + \phi(b)$
  3. $\phi(ab) = \phi(a) \cdot \phi(b)$
- Natural homomorphism between a ring R and its quotient R/I

$$\phi : R \longrightarrow R/I$$

$$a \longmapsto \bar{a}$$

Ring Homomorphisms

- A *homomorphism* between rings R, S is a map $\phi : R \to S$ *preserving the ring structure*
    1. $\phi(1) = 1$
    2. $\phi(a + b) = \phi(a) + \phi(b)$
    3. $\phi(ab) = \phi(a) \cdot \phi(b)$
- Natural homomorphism between a ring R and its quotient R/I
- Two rings R, S are *isomorphic*, denoted $R \simeq S$ if there are two homomorphisms $\phi : R \to S$ and $\psi : S \to R$ such that

$$\phi \circ \psi : S \to S \qquad \text{and} \qquad \psi \circ \phi : R \to R$$

are the *identity* homomorphisms.

$\phi \circ \psi = id_S$

$\psi \circ \phi = id_R$

$\phi \circ \psi(a) = a$

Ring Homomorphisms

- A *homomorphism* between rings R, S is a map $\phi : R \to S$ *preserving the ring structure*
  1. $\phi(1) = 1$
  2. $\phi(a + b) = \phi(a) + \phi(b)$
  3. $\phi(ab) = \phi(a) \cdot \phi(b)$
- Natural homomorphism between a ring R and its quotient R/I
- Two rings R, S are *isomorphic*, denoted $R \simeq S$ if there are two homomorphisms $\phi : R \to S$ and $\psi : S \to R$ such that

$$\phi \circ \psi : S \to S \qquad \text{and} \qquad \psi \circ \phi : R \to R$$

  are the *identity* homomorphisms.
- Example:

$$\mathbb{Z}_6 \simeq \mathbb{Z}_2 \times \mathbb{Z}_3$$

This is particular case of the Chinese Remainder thm

- Background on Rings and Quotients
- Chinese Remainder Theorem
- Variants on Chinese Remainder Theorem
- Conclusion
- Acknowledgements

Chinese Remainder Theorem

- **Setup**: let R be *Euclidean Domain* and $m_1, \ldots, m_s \in R$ be *pairwise coprime*, i.e. $\gcd(m_i, m_j) = 1$, for $i \neq j$. Let $m = m_1 \cdots m_s$.

Chinese Remainder Theorem

- **Setup**: let R be *Euclidean Domain* and $m_1, \ldots, m_s \in R$ be *pairwise coprime*, i.e. $\gcd(m_i, m_j) = 1$, for $i \neq j$. Let $m = m_1 \cdots m_s$.
- *Chinese Remainder Theorem*

$$R/(m) \simeq R/(m_1) \times \cdots \times R/(m_s)$$

Chinese Remainder Theorem

- **Setup**: let R be *Euclidean Domain* and $m_1, \ldots, m_s \in R$ be *pairwise coprime*, i.e. $\gcd(m_i, m_j) = 1$, for $i \neq j$. Let $m = m_1 \cdots m_s$.
- *Chinese Remainder Theorem*

$$R/(m) \simeq R/(m_1) \times \cdots \times R/(m_s)$$

- Example when $R = \mathbb{Z}$: $m = 15$, $m_1 = 3, m_2 = 5$

$$\mathbb{Z}_{15} \simeq \mathbb{Z}_3 \times \mathbb{Z}_5$$

with homomorphisms:

$$a \mod 15 \to (a \mod 3, \; a \mod 5)$$

and

a mod 15

$$(x \mod 3, \; y \mod 5) \to 6 \cdot y - 5 \cdot x \mod 15$$

$(a, a) \longmapsto 6 \cdot a - 5 \cdot a = a \mod 15$

$(x, y) \longmapsto 6 \cdot y - 5x \longmapsto (-5x, y) = (x, y)$

Chinese Remainder Theorem

- **Setup**: let R be *Euclidean Domain* and $m_1, \ldots, m_s \in R$ be *pairwise coprime*, i.e. $\gcd(m_i, m_j) = 1$, for $i \neq j$. Let $m = m_1 \cdots m_s$.
- *Chinese Remainder Theorem*

$$R/(m) \simeq R/(m_1) \times \cdots \times R/(m_s)$$

- Example when $R = \mathbb{Z}$: $m = 15$, $m_1 = 3, m_2 = 5$

$$\underbrace{\mathbb{Z}_{15}}_{\text{big}} \simeq \underbrace{\mathbb{Z}_3 \times \mathbb{Z}_5}_{\text{small rings}}$$

with homomorphisms:

$$a \mod 15 \to (a \mod 3, \ \ a \mod 5)$$

and

$$(x \mod 3, \ \ y \mod 5) \to 6 \cdot y - 5 \cdot x \mod 15$$

- Because it is an isomorphism, can perform *computations* with either representation!

working over small rings can save computational resources!

Chinese Remainder Theorem

- **Setup**: let R be *Euclidean Domain* and $m_1, \ldots, m_s \in R$ be *pairwise coprime*, i.e. $\gcd(m_i, m_j) = 1$, for $i \neq j$. Let $m = m_1 \cdots m_s$.
- *Chinese Remainder Theorem*

$$R/(m) \simeq R/(m_1) \times \cdots \times R/(m_s)$$

- Example when $R = \mathbb{Z}$: $m = 15$, $m_1 = 3, m_2 = 5$

$$\mathbb{Z}_{15} \simeq \mathbb{Z}_3 \times \mathbb{Z}_5$$

  with homomorphisms:

$$a \mod 15 \to (a \mod 3, \quad a \mod 5)$$

  and

$$(x \mod 3, \quad y \mod 5) \to 6 \cdot y - 5 \cdot x \mod 15$$

- Because it is an isomorphism, can perform *computations* with either representation!
- How to prove this theorem? And why is it useful to have this isomorphism? *modular algorithms*!

Chinese Remainder Theorem - Proof for $R = \mathbb{Z}$

- **Setup**: $m_1, \ldots, m_s \in \mathbb{Z}$ be *pairwise coprime*, i.e. $\gcd(m_i, m_j) = 1$, for $i \neq j$. Let $m = m_1 \cdots m_s$.
- *Chinese Remainder Theorem*

$$\mathbb{Z}/(m) \simeq \mathbb{Z}/(m_1) \times \cdots \times \mathbb{Z}/(m_s)$$

Chinese Remainder Theorem - Proof for $R = \mathbb{Z}$

- **Setup**: $m_1, \ldots, m_s \in \mathbb{Z}$ be *pairwise coprime*, i.e. $\gcd(m_i, m_j) = 1$, for $i \neq j$. Let $m = m_1 \cdots m_s$.
- *Chinese Remainder Theorem*

$$\mathbb{Z}/(m) \simeq \mathbb{Z}/(m_1) \times \cdots \times \mathbb{Z}/(m_s)$$

Chinese Remainder Theorem - Proof for $R = \mathbb{Z}$

- **Setup**: $m_1, \ldots, m_s \in \mathbb{Z}$ be *pairwise coprime*, i.e. $\gcd(m_i, m_j) = 1$, for $i \neq j$. Let $m = m_1 \cdots m_s$.
- *Chinese Remainder Theorem*

$$\mathbb{Z}/(m) \simeq \mathbb{Z}/(m_1) \times \cdots \times \mathbb{Z}/(m_s)$$

- One homomorphism is easy:

$$a \mod m \to (a \mod m_1, \ \ldots, \ a \mod m_s)$$

Chinese Remainder Theorem - Proof for $R = \mathbb{Z}$

- **Setup**: $m_1, \ldots, m_s \in \mathbb{Z}$ be *pairwise coprime*, i.e. $\gcd(m_i, m_j) = 1$, for $i \neq j$. Let $m = m_1 \cdots m_s$.
- *Chinese Remainder Theorem*

$$\mathbb{Z}/(m) \simeq \mathbb{Z}/(m_1) \times \cdots \times \mathbb{Z}/(m_s)$$

- One homomorphism is easy:

$$a \mod m \to (a \mod m_1, \ \ldots, \ a \mod m_s)$$

- How can we compute the other homomorphism?
  1. Idea is similar to Lagrange interpolation!

Chinese Remainder Theorem - Proof for $R = \mathbb{Z}$

- **Setup**: $m_1, \ldots, m_s \in \mathbb{Z}$ be *pairwise coprime*, i.e. $\gcd(m_i, m_j) = 1$, for $i \neq j$. Let $m = m_1 \cdots m_s$.
- *Chinese Remainder Theorem*

$$\mathbb{Z}/(m) \simeq \mathbb{Z}/(m_1) \times \cdots \times \mathbb{Z}/(m_s)$$

- One homomorphism is easy:

$$a \mod m \to (a \mod m_1, \ \ldots, \ a \mod m_s)$$

- How can we compute the other homomorphism?
  1. Idea is similar to Lagrange interpolation!
  2. Find elements $L_i \in \mathbb{Z}_m$ such that

$$L_i \equiv \delta_{ij} \mod m_j$$

$$\begin{cases} 1 \mod m_i \\ 0 \mod m_j \quad j \neq i \end{cases}$$

Chinese Remainder Theorem - Proof for $R = \mathbb{Z}$

$\mathbb{Q}[x]$

- **Setup**: $m_1, \ldots, m_s \in \mathbb{Z}$ be *pairwise coprime*, i.e. $\gcd(m_i, m_j) = 1$, for $i \neq j$. Let $m = m_1 \cdots m_s$.
- *Chinese Remainder Theorem*

$$\mathbb{Z}/(m) \simeq \mathbb{Z}/(m_1) \times \cdots \times \mathbb{Z}/(m_s)$$

- One homomorphism is easy:

$$a \mod m \to (a \mod m_1, \ \ldots, \ a \mod m_s)$$

- How can we compute the other homomorphism?
  1. Idea is similar to Lagrange interpolation!
  2. Find elements $L_i \in \mathbb{Z}_m$ such that

$$L_i \equiv \delta_{ij} \mod m_j$$

  3. Then we have

$$(u_1 \mod m_1, \ \ldots, \ u_s \mod m_s) \to u_1 L_1 + \cdots + u_s L_s \mod m$$

  is the other homomorphism

Finding the interpolators L_i

- This part follows from the fact that m_i's are pairwise coprime.

Finding the interpolators L_i

- This part follows from the fact that m_i's are pairwise coprime.
- $\gcd(m_i, m/m_i) = 1 \;\Rightarrow\; \exists\;\; s_i, t_i \in \mathbb{Z}$ s.t.

$$s_i \cdot m_i + t_i \cdot m/m_i = 1$$

$$\prod_{j \neq i} m_j$$

m_i and m/m_i also pairwise coprime

$$\frac{\prod_{j \neq i} (x - \alpha_j)}{\prod_{j \neq i} (\alpha_i - \alpha_j)}$$

Finding the interpolators L_i

- This part follows from the fact that m_i's are pairwise coprime.
- $\gcd(m_i, m/m_i) = 1 \;\Rightarrow\; \exists\;\; s_i, t_i \in \mathbb{Z}$ s.t.

$$s_i \cdot m_i + t_i \cdot m/m_i = 1$$

- Taking $L_i = t_i \cdot m/m_i$ solves this part.

$$L_i = t_i \cdot m/m_i = t_i \cdot \prod_{j \neq i} m_j$$

$$m_j \mid L_i \Rightarrow L_i \equiv 0 \mod m_j \quad (j \neq i)$$

$$L_i = t_i \cdot \frac{m}{m_i} = 1 - s_i m_i \equiv 1 \mod m_i$$

Finding the interpolators L_i

- This part follows from the fact that m_i's are pairwise coprime.
- $\gcd(m_i, m/m_i) = 1 \;\Rightarrow\; \exists \;\; s_i, t_i \in \mathbb{Z}$ s.t.

$$s_i \cdot m_i + t_i \cdot m/m_i = 1$$

- Taking $L_i = t_i \cdot m/m_i$ solves this part.
- This is what we did in our earlier example!

$$\mathbb{Z}_{15} \simeq \mathbb{Z}_3 \times \mathbb{Z}_5$$

$$\gcd(3, 5) = 1$$

$$2 \cdot 3 + (-1) \cdot 5 = 1$$

$L_1 = -5$

$L_2 = 6$

$$\boxed{6y - 5x}$$

Complexity of Computing Homomorphisms

- To compute the first homomorphism, we simply need to compute $a \bmod m_i$ for each m_i, which takes $O(\underbrace{\log m \cdot \log m_i}_{\text{division w/ remainder}})$

$$c \cdot \sum_{i=1}^{s} \log m \cdot \log m_i =$$

$$= c \log m \underbrace{\sum_{i=1}^{s} \log m_i}_{\log\left(\prod_{i=1}^{s} m_i\right) = \log m}$$

$$= c \log^2 m = O(\log^2 m)$$

Complexity of Computing Homomorphisms

- To compute the first homomorphism, we simply need to compute $a \mod m_i$ for each m_i, which takes $O(\log m \cdot \log m_i)$
- Computing second homomorphism:
  - **input:** $(u_1, \ldots, u_s) \in \mathbb{Z}_{m_1} \times \cdots \times \mathbb{Z}_{m_s}$
  - **output:** $a \in \mathbb{Z}_m$ such that $a = u_i \mod m_i$ $i \in [s]$

Complexity of Computing Homomorphisms

- To compute the first homomorphism, we simply need to compute $a \mod m_i$ for each m_i, which takes $O(\log m \cdot \log m_i)$
- Computing second homomorphism:
  - **input:** $(u_1, \ldots, u_s) \in \mathbb{Z}_{m_1} \times \cdots \times \mathbb{Z}_{m_s}$
  - **output:** $a \in \mathbb{Z}_m$ such that $a = u_i \mod m_i$
- By previous slide, enough to compute L_i's

Complexity of Computing Homomorphisms

- To compute the first homomorphism, we simply need to compute $a \mod m_i$ for each m_i, which takes $O(\log m \cdot \log m_i)$
- Computing second homomorphism:
  - **input:** $(u_1, \ldots, u_s) \in \mathbb{Z}_{m_1} \times \cdots \times \mathbb{Z}_{m_s}$
  - **output:** $a \in \mathbb{Z}_m$ such that $a = u_i \mod m_i$
- By previous slide, enough to compute L_i's
- First, need to compute m (as we are only given m_i's as input). We assume here $m_i \geq 2$

Complexity of Computing Homomorphisms

- To compute the first homomorphism, we simply need to compute $a \mod m_i$ for each m_i, which takes $O(\log m \cdot \log m_i)$
- Computing second homomorphism:
  - **input:** $(u_1, \ldots, u_s) \in \mathbb{Z}_{m_1} \times \cdots \times \mathbb{Z}_{m_s}$
  - **output:** $a \in \mathbb{Z}_m$ such that $a = u_i \mod m_i$
- By previous slide, enough to compute L_i's
- First, need to compute m (as we are only given m_i's as input). We assume here $m_i \geq 2$ $O(\log m_1 \cdot \log m_2)$
- Computing $m_1 m_2$, then $m_1 m_2 m_3$, until we compute m, we have:

$$c \cdot \sum_{i=2}^{s} \underbrace{\overbrace{\log(m_1 \cdots m_{i-1})}^{\leq \log m}}_{\text{computes } m_1 \cdots m_i} \cdot \log m_i \leq c \cdot \log(m) \cdot \underbrace{\sum_{i=2}^{s} \log m_i}_{\log\left(\prod_{i=2}^{s} m_i\right) \leq \log(m)} \leq c \cdot (\log m)^2$$

$O(\log^2 m)$ operations.

Complexity of Computing Homomorphisms

- To compute the first homomorphism, we simply need to compute $a \mod m_i$ for each m_i, which takes $O(\log m \cdot \log m_i)$
- Computing second homomorphism:
  - **input:** $(u_1, \ldots, u_s) \in \mathbb{Z}_{m_1} \times \cdots \times \mathbb{Z}_{m_s}$
  - **output:** $a \in \mathbb{Z}_m$ such that $a = u_i \mod m_i$
- By previous slide, enough to compute L_i's
- First, need to compute m (as we are only given m_i's as input). We assume here $m_i \geq 2$
- Computing $m_1 m_2$, then $m_1 m_2 m_3$, until we compute m, we have:

$$c \cdot \sum_{i=2}^{s} \log(m_1 \cdots m_{i-1}) \cdot \log m_i \leq c \cdot \log(m) \cdot \sum_{i=2}^{s} \log m_i \leq c \cdot (\log m)^2$$

- Now we can compute each element $\boxed{m/m_i}$ by our division algorithm in $O(\underline{\log(m)}\,\underline{\log(m_i)})$ time

$O(\log m_i \cdot \log(m/m_i))$

Complexity of Computing Homomorphisms

- Now we have computed $m, m/m_1, \ldots, m/m_s$ in time $O(\log^2 m)$ ops

Complexity of Computing Homomorphisms

- Now we have computed $m, m/m_1, \ldots, m/m_s$ in time $O(\log^2 m)$ ops
- What is left is to compute the interpolators L_i's

Complexity of Computing Homomorphisms

- Now we have computed $m, m/m_1, \ldots, m/m_s$ in time $O(\log^2 m)$ ops
- What is left is to compute the interpolators L_i's
- We know that $L_i = t_i \cdot m/m_i$, where

$$s_i m_i + t_i m/m_i = 1$$

Extended Euclidean Algorithm

Complexity of Computing Homomorphisms

- Now we have computed $m, m/m_1, \ldots, m/m_s$ in time $O(\log^2 m)$ ops
- What is left is to compute the interpolators L_i's
- We know that $L_i = t_i \cdot m/m_i$, where

$$s_i m_i + t_i m/m_i = 1$$

- Thus, we need the extended Euclidean algorithm to compute (s_i, t_i)

Complexity of Computing Homomorphisms

- Now we have computed $m, m/m_1, \ldots, m/m_s$ in time $O(\log^2 m)$ ops
- What is left is to compute the interpolators L_i's
- We know that $L_i = t_i \cdot m/m_i$, where

$$s_i \underline{m_i} + t_i \underline{m/m_i} = 1$$

- Thus, we need the extended Euclidean algorithm to compute (s_i, t_i)
- From previous class, cost is $O(\log(m/m_i) \cdot \log(m_i))$

Complexity of Computing Homomorphisms

- Now we have computed $m, m/m_1, \ldots, m/m_s$ in time $O(\log^2 m)$ ops
- What is left is to compute the interpolators L_i's
- We know that $L_i = t_i \cdot m/m_i$, where

$$s_i m_i + t_i m/m_i = 1$$

- Thus, we need the extended Euclidean algorithm to compute (s_i, t_i)
- From previous class, cost is $O(\log(m/m_i) \cdot \log(m_i))$
- Gives total running time of $O(\log^2 m)$

$$(u_1, \ldots, u_s) \longmapsto u_1 L_1 + \cdots u_s L_s \mod m$$

Complexity of Computing Homomorphisms

- Now we have computed $m, m/m_1, \ldots, m/m_s$ in time $O(\log^2 m)$ ops
- What is left is to compute the interpolators L_i's
- We know that $L_i = t_i \cdot m/m_i$, where

$$s_i m_i + t_i m/m_i = 1$$

- Thus, we need the extended Euclidean algorithm to compute (s_i, t_i)
- From previous class, cost is $O(\log(m/m_i) \cdot \log(m_i))$
- Gives total running time of $O(\log^2 m)$

Both homomorphisms can be computed with $O(\log^2 m)$ operations.

Mixed Radix Representation

- **Setup:** $0 \leq a < m = m_1 \cdots m_s$, where the $m_i \geq 2$ are integers which *are not necessarily coprime*

Mixed Radix Representation

- **Setup:** $0 \leq a < m = m_1 \cdots m_s$, where the $m_i \geq 2$ are integers which *are not necessarily coprime*
- *Theorem*: Can write a uniquely as

$$a = a_0 + a_1 \cdot m_1 + a_2 \cdot m_1 m_2 + \cdots + a_{s-1} \cdot m_1 m_1 \cdots m_{s-1}$$

$$(a_0, a_1, a_2, \ldots, a_{s-1})$$

$$a_0 \in \mathbb{Z}_{m_1}$$

$$a_1 \in \mathbb{Z}_{m_2}$$

$$a_i \in \mathbb{Z}_{m_{i+1}}$$

Mixed Radix Representation

- **Setup:** $0 \leq a < m = m_1 \cdots m_s$, where the $m_i \geq 2$ are integers which *are not necessarily coprime*
- *Theorem*: Can write a uniquely as

$$a = a_0 + a_1 \cdot m_1 + a_2 \cdot m_1 m_2 + \cdots + a_{s-1} \cdot m_1 m_1 \cdots m_{s-1}$$

- Proof by induction
  1. Base case: $s = 1$ $\quad a = a_0 \mod m_1$

Mixed Radix Representation

$$a = (a_0, \dots, a_{s-2}, b)$$
$\hookrightarrow a_{s-1}$

- **Setup:** $0 \leq a < m = m_1 \cdots m_s$, where the $m_i \geq 2$ are integers which *are not necessarily coprime*
- *Theorem*: Can write a uniquely as

$$a = a_0 + a_1 \cdot m_1 + a_2 \cdot m_1 m_2 + \cdots + a_{s-1} \cdot m_1 m_1 \cdots m_{s-1}$$

- Proof by induction
  1. Base case: $s = 1$
  2. Assuming we know for $s - 1$ numbers $m_1, \dots, m_{s-1}$

$a = a' \bmod m_1 m_2 \cdots m_{s-1}$ $\quad \underline{a} = \underline{b} \cdot \underline{m_1 \cdots m_{s-1}} + \underline{a'}$

induction hypothesis a' uniquely $(a_0, \dots, a_{s-2})$

$0 \leq a < m \Rightarrow b \in \mathbb{Z}_{m_s}$ and b unique

Incremental Chinese Remaindering

- **Setup:** here we are back to the setup that $\underbrace{\gcd(m_i, m_j) = 1}_{\text{coprime}}$ (the CRT setup)
- Incremental Chinese remaindering computes

$$\underline{(a \mod m_1)}, \quad \underline{(a \mod m_1 m_2)}, \quad \cdots, \quad \underline{(a \mod m_1 m_2 \cdots m_{s-1})}$$

Incremental Chinese Remaindering

- **Setup:** here we are back to the setup that $\gcd(m_i, m_j) = 1$ (the CRT setup)
- Incremental Chinese remaindering computes

$$(a \mod m_1), \quad (a \mod m_1 m_2), \quad \cdots, \quad (a \mod m_1 m_2 \cdots m_{s-1})$$

- Why would we want to do that?
  - in some applications, we sometimes do not know in advance how big the output integer will be

Incremental Chinese Remaindering

$a > p_1 p_2 p_3$ we need another prime

- **Setup:** here we are back to the setup that $\gcd(m_i, m_j) = 1$ (the CRT setup)
- Incremental Chinese remaindering computes

$$(a \mod m_1), \quad (a \mod m_1 m_2), \quad \cdots, \quad (a \mod m_1 m_2 \cdots m_{s-1})$$

- Why would we want to do that?
  - in some applications, we sometimes do not know in advance how big the output integer will be
  - thus, we compute the result modulo many primes (which we have to decide "on the fly")

Incremental Chinese Remaindering

- **Setup:** here we are back to the setup that $\gcd(m_i, m_j) = 1$ (the CRT setup)
- Incremental Chinese remaindering computes

$$(a \mod m_1), \quad (a \mod m_1 m_2), \quad \cdots, \quad (a \mod m_1 m_2 \cdots m_{s-1})$$

- Why would we want to do that?
  - in some applications, we sometimes do not know in advance how big the output integer will be
  - thus, we compute the result modulo many primes (which we have to decide "on the fly")
  - if we get same number modulo $p_1 p_2 \cdots p_k$ for some value of k, we "guess" that we have the right result.

$a < p_1 p_2 p_3$ $\qquad$ $a \equiv a \mod p_1 p_2 p_3 p_4$

Incremental Chinese Remaindering

- **Setup:** here we are back to the setup that $\gcd(m_i, m_j) = 1$ (the CRT setup)
- Incremental Chinese remaindering computes

$$(a \mod m_1), \quad (a \mod m_1 m_2), \quad \cdots, \quad (a \mod m_1 m_2 \cdots m_{s-1})$$

- Why would we want to do that?
  - in some applications, we sometimes do not know in advance how big the output integer will be
  - thus, we compute the result modulo many primes (which we have to decide "on the fly")
  - if we get same number modulo $p_1 p_2 \cdots p_k$ for some value of k, we "guess" that we have the right result.
  - Good for randomized algorithms

Conclusion

In today's lecture, we learned

- Properties of Rings and its quotients
- Chinese Remainder Theorem (CRT)
- Analysis of computation of homomorphisms in CRT
- Mixed radix representation (alternative to CRT)
- Iterative CRT and how one could use it to develop randomized algorithms with lower bit complexity

Acknowledgement

- Based largely on Arne's notes

  `https://cs.uwaterloo.ca/~r5olivei/courses/2021-winter-cs487/lec6-ref.pdf`