WEB3CLUBS FOUNDATION LIMITED

Course Instructor: DR. Cyprian Omukhwaya Sakwa
PHONE: +254723584205 Email: cypriansakwa@gmail.com

Foundational Mathematics for Web3 Builders

Implemented in RUST

Lecture 44

August 19, 2024

Order of an element of a direct product group

The order of an element in a direct product group of a finite number of finite groups is the least common multiple (lcm) of the orders of the components of the elements.
Given components $a_1, a_2, \cdots, a_n$ in a direct product group, we have $(a_1, a_2, \cdots, a_n)| = \mathrm{lcm}\big(|a_1|, |a_2|, \cdots, |a_n|\big).$

Theorem 17

Let $G \times H$ be a group. The order of an element $(a, b) \in G \times H$ is the lcm of the orders of a and b. That is, $|(a, b)| = \mathit{lcm}\big(|a|, |b|\big)$.

Example 82

Find the order of $(1,1) \in (\mathbb{Z}_3 \times \mathbb{Z}_9)$.

Solution

First, $\mid 1 \mid \in \mathbb{Z}_3 = 3$ and $\mid 1 \mid \in \mathbb{Z}_9 = 9$.

lcm$(3,9) = 9$. Thus the order of $(1,1)$ is 9.

We would have obtained the same result by direct calculation as follows;

$1(1,1) = (1,1),\ 2(1,1) = (2,2),\ 3(1,1) = (0,3),\ 4(1,1) = (1,4),$

$5(1,1) = (2,5),\ 6(1,1) = (0,6),\ 7(1,1) = (1,7),\ 8(1,1) = (2,8),$

$9(1,1) = (0,0)$. This gives $|(1,1)| = 9$.

Example 83

Find the order of $(1,1) \in (\mathbb{Z}_7 \times \mathbb{Z}_{10})$.

Solution

lcm$(7,10) = 70$. Thus the order of $(1,1)$ is 70.

Let $G \times H$ be a group for any groups G and H. To find the order of $(x,y) \in G \times H$ proceed as follows. First find the order of x in G and the order of y in H. Let's say that $|x| = n$ and $|y| = m$, then the order of (x,y) is given by the lcm of n and m.

Example 84

Find the order of $(5,6) \in (\mathbb{Z}_7 \times \mathbb{Z}_{10})$.

Solution

From Examples 75 and 76 above, $|5| \in \mathbb{Z}_7 = 7$ and $|6| \in \mathbb{Z}_{10} = 5$. Thus $|(5,6)|$ is the lcm of 7 and 5 which is 35.

Example 85

Find the order of $(6,9) \in (\mathbb{Z}_8 \times \mathbb{Z}_{12})$.

Solution

Here, $|6| \in \mathbb{Z}_8 = 4$ and $|9| \in \mathbb{Z}_{12} = 4$. Thus $|(6,9)|$ is the lcm of 4 and 4 which is 4.

Example 86

Find the order of $(2,8,6) \in (\mathbb{Z}_6 \times \mathbb{Z}_{13} \times \mathbb{Z}_9)$.

Solution

We find $|2| \in \mathbb{Z}_6$.

$1 \cdot 2 = 2,\ 2 \cdot 2 = 4,\ 3 \cdot 2 = 0$. Thus $|2| = 3$.

We find $|8| \in \mathbb{Z}_{13}$.

$1 \cdot 8 = 8,\ 2 \cdot 8 = 3,\ 3 \cdot 8 = 11,\ 4 \cdot 8 = 6,\ 5 \cdot 8 = 1,\ 6 \cdot 8 = 9,\ 7 \cdot 8 = 4,$

$8 \cdot 8 = 12,\ 9 \cdot 8 = 7,\ 10 \cdot 8 = 2,\ 11 \cdot 8 = 10,\ 12 \cdot 8 = 5,\ 13 \cdot 8 = 0.$

Thus $|8| = 13$.

Solution (conti...)

Now find $|6| \in \mathbb{Z}_9$.

$1 \cdot 6 = 6,\ 2 \cdot 6 = 3,\ 3 \cdot 6 = 0$. Thus $|6| = 3$.

Thus, order of $(2, 8, 6) \in (\mathbb{Z}_6 \times \mathbb{Z}_{13} \times \mathbb{Z}_9) = \text{lcm}(3, 13, 3) = 39$.

Example 87

Find the order of $(6, 7) \in (\mathbb{Z}_8 \times \mathbb{Z}^*_{18})$

Solution

$|\ 6\ | \in \mathbb{Z}_8 = 4$ and $|\ 7\ | \in \mathbb{Z}^*_{12} = 3$.

Thus, $|\ (6, 7)\ | \in (\mathbb{Z}_8 \times \mathbb{Z}^*_{18}) = \text{lcm}(4, 3) = 12$ 5(6,7 =

12

To confirm this, we have $(12 \cdot 6, 7^{12}) = (0, 1)$ the identity element of the group.

Example 88

Find the order of $(16, 43) \in (\mathbb{Z}_{23}^* \times \mathbb{Z}_{52}^*)$

Solution

$| 16 | \in \mathbb{Z}_{23}^* = 11$ and $| 43 | \in \mathbb{Z}_{52}^* = 6.$
Thus, $| (16, 43) | \in (\mathbb{Z}_{23}^* \times \mathbb{Z}_{52}^*) = \text{lcm}(11, 6) = 66$

To confirm this, we have $(16^{66}, 43^{66}) = (1, 1)$ the identity element of the group.

The following Rust program computes the order of an element $(a, b) \in \mathbb{Z}_n \times \in \mathbb{Z}_m$, where $\mathbb{Z}_n$ and $\mathbb{Z}_m$ are additive groups. Recall that the order of an element (a, b) in this direct product is the least common multiple of the orders of $a \in \mathbb{Z}_n$ and $b \in \mathbb{Z}_m$. This order is the smallest positive integer k such that $(ka, kb) = (0, 0)$.

```rust
// Function to compute the greatest common divisor
fn gcd(a: u64, b: u64) -> u64 {
        if b == 0 {
                a
        } else {
                gcd(b, a % b)
        }
}

// Function to compute the least common multiple
fn lcm(a: u64, b: u64) -> u64 {
        (a * b) / gcd(a, b)
}

// Function to compute the additive order of a in Z_n
fn additive_order(a: u64, n: u64) -> u64 {
        n / gcd(a, n)
}

// Function to compute the order of (a, b) in Z_n x Z_m
fn order(a: u64, n: u64, b: u64, m: u64) -> u64 {
        let add_order_a = additive_order(a, n);
        let add_order_b = additive_order(b, m);
        lcm(add_order_a, add_order_b)
}

fn main() {
        let a = 5;
        let n = 7;
        let b = 6;
        let m = 10;
```

```
        let result = order(a, n, b, m);
        println!("The␣order␣of␣({},␣{})␣in␣Z_{}␣x␣Z_{}␣is␣{}", a, b, n, m, result);
}
```

Understanding the Rust code

1. Greatest Common Divisor (gcd): The gcd function computes the greatest common divisor (gcd) of two numbers a and b using the Euclidean algorithm.

```
fn gcd(a: u64, b: u64) -> u64 {
        if b == 0 {
                a
        } else {
                gcd(b, a % b)
        }
}
```

- The function repeatedly replaces a with b and b with $a\%b$ until b becomes 0, at which point a is the $\gcd$.

Understanding the Rust code (conti...)

(2) Least Common Multiple (lcm): The lcm function calculates the least common multiple (lcm) of two numbers a and b.

```
fn lcm(a: u64, b: u64) -> u64 {
        (a * b) / gcd(a, b)
}
```

- The least common multiple of a and b is calculated using the formula lcm $= \frac{a \times b}{\gcd(a,b)}$. ✓

(3) Additive Order in $\mathbb{Z}_n$: This function computes the order of an element a in the group $\mathbb{Z}_n$.

```
fn additive_order(a: u64, n: u64) -> u64 {
        n / gcd(a, n)
}
```

- The order of $a \in \mathbb{Z}_n$ is calculated using the formula $\frac{n}{\gcd(a,n)}$.

Understanding the Rust code (conti...)

(4) Order in $\mathbb{Z}_n \times \mathbb{Z}_m$: This function computes the order of the pair $(a, b) \in \mathbb{Z}_n \times \in \mathbb{Z}_m$.

```rust
fn order(a: u64, n: u64, b: u64, m: u64) -> u64 {
        let add_order_a = additive_order(a, n);
        let add_order_b = additive_order(b, m);
        lcm(add_order_a, add_order_b)
}
```

- The order of $(a, b) \in \mathbb{Z}_n \times \in \mathbb{Z}_m$ is the least common multiple of the additive orders of $a \in \mathbb{Z}_n$ and $b \in \mathbb{Z}_m$.

(5) Main Function: This function initializes some example values and calls the order function.

The following Rust program computes the order of an element $(a, b) \in \mathbb{Z}_n^* \times \mathbb{Z}_m^*$, where Z_n^* and Z_m^* are multiplicative groups. The order of an element (a, b) in this direct product is the least common multiple of the orders of $a \in \mathbb{Z}_n^*$ and $b \in \mathbb{Z}_m^*$. This order is the smallest positive integer k such that $(a^k, b^k) = (1, 1)$.

```
use num::integer::lcm;

// Function to compute the multiplicative order of a modulo n
fn multiplicative_order(a: u64, n: u64) -> u64 {
        let mut order = 1;
        let mut power = a % n;
        while power != 1 {
                power = (power * a) % n;
                order += 1;
        }
        order
}

// Function to compute the order of (a, b) in Z_n^* x Z_m^*
fn order(a: u64, n: u64, b: u64, m: u64) -> u64 {
        let order_a = multiplicative_order(a, n);
        let order_b = multiplicative_order(b, m);
        lcm(order_a, order_b)
}

```

$3 \in \mathbb{Z}_5^\times$

let order = 1

let power = 3

power = (3 × 3) % 5

= 4

order = 2

power = (4 × 3) % 5

= 2

order = 3

```
fn main() {
        let a = 16;
        let n = 23;
        let b = 43;
        let m = 52;

        let result = order(a, n, b, m);
println!("The␣order␣of␣({},␣{})␣in␣Z_{}^*␣x␣Z_{}^*␣is␣{}", a, b, n, m, result);
}
```

Understanding the Rust code

1. multiplicative_order $(a : u64, n : u64) -> u64$: This function calculates the multiplicative order of an element a modulo n.

```
fn multiplicative_order(a: u64, n: u64) -> u64 {
        let mut order = 1;
        let mut power = a % n;
        while power != 1 {
                power = (power * a) % n;
                order += 1;
        }
        order
}
```

- Initialization:
  - order is set to 1. This variable keeps track of the current order
  - power is initialized to $a\%n$. This represents the current power of a modulo n.

Understanding the Rust code (conti...)

- Loop:
  - While power is not equal to 1, the function repeatedly multiplies power by a and reduces modulo n.
  - Each iteration increments the order
- Termination: The loop terminates when $a^k \equiv 1 \bmod n$, and the function returns the order.

(2) order $(a : u64, n : u64, b : u64, m : u64) -> u64$: This function calculates the order of the pair (a, b) in $\mathbb{Z}_n^* \times \mathbb{Z}_m^*$.

```
fn order(a: u64, n: u64, b: u64, m: u64) -> u64 {
        let order_a = multiplicative_order(a, n);
        let order_b = multiplicative_order(b, m);
        lcm(order_a, order_b)
}
```

Understanding the Rust code (conti...)

- Order Calculation:
  - It computes the multiplicative order of a modulo n and b modulo m using the multiplicative_order function.
  - It then computes the least common multiple (lcm) of these two orders using the lcm function from the num crate.

Exercise 6

1. Let $G = \{a, b, c, d\}$ with multiplication and addition tables defined by table 5

$\cdot$	a	b	c	d
a	c	a	d	b
b	a	b	c	d
c	d	c	b	a
d	b	d	a	c

a: Multiplication table for G

$+$	a	b	c	d
a	a	b	c	d
b	b	c	d	a
c	c	d	b	a
d	d	b	c	c

b: Addition table for G

Table 5

Are $(G, \cdot)$ and $(G, +)$ groups? Explain.

2. In $\mathbb{Z}_7$, find the following;

a) $-2,\ -3,\ -4,\ -6$

b) $1/5,\ 2/5,\ 4/5,\ 3/6$

Exercise (conti...)

3. Does $\mathbb{Z}_3$, the set of residue classes modulo 3 form a group

   a) Under addition?

   b) Under multiplication? Show your working.

4. Do nonzero residue classes modulo 3 form a group under multiplication? Show your working.
5. Do nonzero residue classes modulo 8 form a group under multiplication? Show your working.

Exercise (conti...)

6. Consider the group $G = \{1, 2, 3, 4\}$ under multiplication modulo 5.

   a) Draw multiplication table of G.

   b) Find 2^{-1}, 3^{-1} and 4^{-1}.

   c) Find $^{2}/_{3}$ and $^{3}/_{4}$.

7. Let $H = \{1, 5, 7, 11, 13, 17\}$ be the reduced system modulo 18. Find multiplication table for H. Does H form a group under multiplication modulo 18? Find the inverses of 7 and 11.

Exercise (conti...)

8. Given $\mathbb{Z}_2 = \{0, 1\}$ and $\mathbb{Z}_5^* = \{1, 2, 3, 4\}$, find;

   a) $\mathbb{Z}_2 \times \mathbb{Z}_5$.

   b) $(1, 1)(1, 4)$

   c) $(0, 4)(1, 2)$

   d) $(1, 3)(1, 3)$

   e) Identity element in $\mathbb{Z}_2 \times \mathbb{Z}_5^*$.

   f) Inverse of $(0, 1)$

   g) Inverse of $(1, 2)$

   h) Inverse of $(1, 4)$

9. Find the order of $(991, 1396)$ in;

   a) $\mathbb{Z}_{1081} \times \mathbb{Z}_{1481}$

   b) $\mathbb{Z}_{1081} \times \mathbb{Z}_{1481}^*$

   c) $\mathbb{Z}_{1081}^* \times \mathbb{Z}_{1481}^*$

   You may use Rust codes.