

# Modular arithmetic

# A Theorem

## Theorem

$$\forall z \in \mathbb{Z} \forall d \in \mathbb{N} \exists! q \in \mathbb{Z} \exists! r \in \mathbb{Z} \ (z = qd + r) \wedge (0 \leq r < d)$$

# A Theorem

## Theorem

$$\forall z \in \mathbb{Z} \forall d \in \mathbb{N} \exists! q \in \mathbb{Z} \exists! r \in \mathbb{Z} (z = qd + r) \wedge (0 \leq r < d)$$

### **Theorem: (The Quotient-Remainder Theorem).**

Given any integer  $z$  and any positive integer  $d$ , there is exactly one way to express  $z$  in the form  $z = qd + r$ , where  $q$  is an integer and  $r \in \{0, 1, \dots, d - 1\}$ .

In the expression  $z = qd + r$ ,  $q$  is called the **quotient** (when  $z$  is divided by  $d$ ) and  $r$  is called the **remainder** (when  $z$  is divided by  $d$ ).

# Picturing $q$ and $r$

Fix a choice of  $z \in \mathbb{Z}$  and  $d \in \mathbb{N}$ .

Now picture a number line with the integers marked in some dramatic way.

Now:  $qd$  is the integer multiple of  $d$  that is closest to  $z$  but NOT to the right of  $z$ ; and  $r$  is the distance between  $qd$  and  $z$ .

A picture will help...

# 'mod' and 'div'

We define:  $q = z \text{ div } d; \quad r = z \text{ mod } d.$

You may like to say that:

- $z \text{ div } d$  gives the **quotient** when  $z$  is divided by  $d$ ;
- $z \text{ mod } d$  gives the **remainder** when  $z$  is divided by  $d$ .

# Examples

Q: Evaluate the following expressions:

$87 \bmod 13$

$-100 \operatorname{div} 13$

# Examples

Q: Evaluate the following expressions:

$$87 \bmod 13$$

$$-100 \operatorname{div} 13$$

A:

$$\text{Since } 87 = 6(13) + 9, 87 \bmod 13 = 9.$$

$$\text{Since } -100 = (-8)(13) + 4, -100 \operatorname{div} 13 = -8.$$

# The division algorithm

The ‘primary school’ method of finding quotient and remainder is to use *repeated subtraction*. This only works for non-negative  $z$ .

**Input:**  $z \in \mathbb{Z}_{\geq 0}$  and  $d \in \mathbb{N}$ .

**Output:**  $q = z \operatorname{div} d$  and  $r = z \bmod d$ .

**Method:**

Set  $r = z$ ,  $q = 0$ .

Loop: If  $r < d$  stop.

    Replace  $r$  by  $r - d$ .

    Replace  $q$  by  $q + 1$ .

Repeat loop



# The division algorithm

The ‘primary school’ method of finding quotient and remainder is to use *repeated subtraction*. This only works for non-negative  $z$ .

**Input:**  $z \in \mathbb{Z}_{\geq 0}$  and  $d \in \mathbb{N}$ .

**Output:**  $q = z \operatorname{div} d$  and  $r = z \bmod d$ .

**Method:**

Set  $r = z$ ,  $q = 0$ .

Loop: If  $r < d$  stop.

Replace  $r$  by  $r - d$ .

Replace  $q$  by  $q + 1$ .

Repeat loop

Q: Some small modifications to the algorithm allow it cope also with negative  $z$ . Could you make them?

# Congruence modulo $d$

Let  $d \in \mathbb{N}$ . The **congruence modulo  $d$**  relation  $R_d \subseteq \mathbb{Z} \times \mathbb{Z}$  is defined by

$$aR_db \Leftrightarrow (\exists k \in \mathbb{Z} \ a = b + kd).$$

We have unusual notation for this relation. We write

$$a \equiv b \pmod{d}$$

to mean  $aR_db$

# Congruence modulo $d$

Let  $d \in \mathbb{N}$ . The **congruence modulo  $d$**  relation  $R_d \subseteq \mathbb{Z} \times \mathbb{Z}$  is defined by

$$aR_db \Leftrightarrow (\exists k \in \mathbb{Z} \ a = b + kd).$$

We have unusual notation for this relation. We write

$$a \equiv b \pmod{d}$$

to mean  $aR_db$

Understanding the relation:

- Two integers are congruent modulo  $d$  IFF their difference is a multiple of  $d$ .
- Two integers are congruent modulo  $d$  IFF they leave the same remainder upon division by  $d$ .

# Proof

Let  $a, b \in \mathbb{Z}$  and let  $d \in \mathbb{N}$ .

# Proof

Let  $a, b \in \mathbb{Z}$  and let  $d \in \mathbb{N}$ . Suppose that  $a \equiv b \pmod{d}$ . Then there exists  $k \in \mathbb{Z}$  such that  $a = b + kd$ . By the Quotient-Remainder Theorem, there exist unique integers  $q_1, r_1$  such that  $b = q_1d + r_1$  and  $0 \leq r_1 < d$ . Then we have

$$a = b + kd = (q_1d + r_1) + kd = (q_1 + k)d + r_1.$$

The uniqueness part of the Quotient-Remainder Theorem gives that  $r_1$  is the remainder when  $a$  is divided by  $d$ . Thus  $a$  and  $b$  leave the same remainder upon division by  $d$ .

# Proof

Let  $a, b \in \mathbb{Z}$  and let  $d \in \mathbb{N}$ . Suppose that  $a \equiv b \pmod{d}$ . Then there exists  $k \in \mathbb{Z}$  such that  $a = b + kd$ . By the Quotient-Remainder Theorem, there exist unique integers  $q_1, r_1$  such that  $b = q_1d + r_1$  and  $0 \leq r_1 < d$ . Then we have

$$a = b + kd = (q_1d + r_1) + kd = (q_1 + k)d + r_1.$$

The uniqueness part of the Quotient-Remainder Theorem gives that  $r_1$  is the remainder when  $a$  is divided by  $d$ . Thus  $a$  and  $b$  leave the same remainder upon division by  $d$ . Now suppose that  $a$  and  $b$  leave the same remainder upon division by  $d$ . Then there exist  $q_1, q_2, r \in \mathbb{Z}$  such that  $a = q_1d + r$  and  $b = q_2d + r$  and  $0 \leq r < d$ . Now  $r = b - q_2d$ , so

$$a = q_1d + r = q_1d + b - q_2d = b + (q_1 - q_2)d.$$

Hence, by definition,  $a \equiv b \pmod{d}$ . □

# Example

Example:  $-17 \equiv 15 \pmod{8}$  since  
 $(-17) - 15 = -32 = (-4)8.$

# $\equiv$ partitions the integers

For any  $d \in \mathbb{N}$  and any  $a \in \mathbb{Z}$  the **congruence class**  $[a]_d$  (or 'equivalence class') of  $a$  modulo  $d$  is defined by

$$[a]_d = \{m \in \mathbb{Z} \mid m \equiv a \pmod{d}\}.$$

**Lemma:**  $R_d$  induces the partition  $\{[0]_d, [1]_d, \dots, [d-1]_d\}$  on  $\mathbb{Z}$

Example:  $\mathcal{P} = \{[0]_5, [1]_5, [2]_5, [3]_5, [4]_5\}$  is a partition of  $\mathbb{Z}$

Q: Can you see how the Q-R theorem can be used to prove the lemma?



# Modular arithmetic

**Theorem:** For any  $a_1, a_2, b_1, b_2 \in \mathbb{Z}$  and  $d \in \mathbb{N}$ :

$$\boxed{\begin{array}{l} a_1 \equiv a_2 \pmod{d} \\ b_1 \equiv b_2 \pmod{d} \end{array}} \implies \boxed{\begin{array}{l} a_1 \pm b_1 \equiv a_2 \pm b_2 \pmod{d} \\ a_1 \times b_1 \equiv a_2 \times b_2 \pmod{d} \end{array}}$$

Example:

Since

$$27 \equiv -1 \pmod{7} \text{ and } 36 \equiv 1 \pmod{7}$$

we have

$$27 + 36 \equiv -1 + 1 \equiv 0 \pmod{7}$$

and

$$27 \times 36 \equiv -1 \times 1 \equiv -1 \pmod{7}$$

# A key idea

When computing “modulo  $d$ ”, you may at any time replace a number by something to which it is equivalent. In this way, you may simplify computations so that you never have to work with large integers.

# A key idea

When computing “modulo  $d$ ”, you may at any time replace a number by something to which it is equivalent. In this way, you may simplify computations so that you never have to work with large integers.

Examples:

$$379 - 803 \equiv 1 - 5 \equiv -4 \equiv 3 \pmod{7}$$

and

$$379 \times 803 \equiv 1 \times 5 \equiv 5 \pmod{7}$$

and

$$803^5 \equiv 5^5 \equiv 25 \times 25 \times 5 \equiv 4 \times 4 \times 5 \equiv 80 \equiv 3 \pmod{7}$$

# An important problem

The following problem is called the **discrete logarithm problem**: Given  $d \in \mathbb{N}$  and  $A, Q \in \{1, \dots, d-1\}$ , find  $x \in \{1, \dots, d-1\}$  such that  $A^x \equiv Q \pmod{d}$ .

A naive solution to the problem: Compute  $A^1 \pmod{d}, A^2 \pmod{d}, \dots$  until one of your computations produces  $Q$ .

A randomised naive solution to the problem: Repeatedly, select a number  $t$  from  $\{1, 2, \dots, d-1\}$  at random and compute  $A^t \pmod{d}$ . Stop when one of your computations produces  $Q$ .

Your privacy on the internet often relies on the following fact: For certain choices of  $d$ ,  $A$  and  $Q$ , the naive solution to the discrete logarithm problem will take a very long time and the randomised naive solution to the discrete logarithm problem will almost certainly take a very long time.