

# Alternative Methods for Calculating Inverses

In class, we saw how to use the **extended gcd algorithm** to build a formal algorithm for **calculating inverses**.

We will now see two alternative ways of calculating inverses; they also start from the **extended gcd algorithm**

The first notes that it is not necessary to use a formal algorithmic approach, we can just *unwind* the equations to find the inverse.

# Idea: “Iterate backwards”:

- Starting with step 0, number steps of Euclidean algorithm.
- The equation at step  $i$ , will be denoted by  $m_i = n_i q_i + r_i$ .
- After carrying out step  $i$  of Euclidean algorithm, transform it into  $r_i = m_i - n_i q_i$ .
- Let  $r_k$  (step  $k$ ) be last non-zero remainder.  
Recall that if  $r_k = 1$ ,  $\Rightarrow n_0$  has an inverse mod  $m_0$   
(If  $r_k \neq 1$ , then  $n_0$  has no inverse mod  $m_0$ )
- Recall that  $m_i = n_{i-1}$  and  $n_i = r_{i-1}$ .
- Iterate backwards starting with

$$\begin{aligned} r_k = 1 &= m_k - n_k q_k = n_{k-1} - r_{k-1} q_k \\ &= n_{k-1} - (m_{k-1} - n_{k-1} q_{k-1}) q_k \\ &= -m_{k-1} q_k + n_{k-1} (1 + q_{k-1} q_k) \dots \end{aligned}$$

## Example:

Find the inverse of 15 mod 26.

Step 0:  $26 = 1(15) + 11$

$$r_0 = 11 = 26 - 1(15)$$

Step 1:  $15 = 1(11) + 4$

$$r_1 = 4 = 15 - 1(11)$$

Step 2:  $11 = 2(4) + 3$

$$r_2 = 3 = 11 - 2(4)$$

Step 3:  $4 = 1(3) + 1$

$$r_3 = 1 = 4 - 1(3)$$

Iterating "backwards" gives:

Step "3":  $1 = 4 - 1(3)$

Step "2":  $1 = 4 - 1(11 - 2(4)) = -1(11) + 3(4)$

Step "1":  $1 = -1(11) + 3(15 - 1(11)) = 3(15) - 4(11)$

Step "0":  $1 = 3(15) - 4(26 - 1(15)) = -4(26) + 7(15)$

So,  $1 = -4(26) + 7(15)$  and 7 is the inverse of 15 mod 26.

The **backwards iteration** technique we just saw is easier to implement “by hand” than the formal technique we learnt in class. In reality, though, it’s essentially the same as the technique we learnt in class.

In both techniques, we first ran the **extended gcd algorithm** until we found that the GCD equals 1, and then ran the second part of the algorithm backwards, using the intermediate results calculated by the initial GCD algorithm

On the next page we will see a **one-stage** method that only runs forwards and not backwards. See if you can figure out why it works.

# Another idea ("simultaneous calculation"):

- Number steps of Euclidean algorithm starting with step 0.
- Quotient obtained at step  $i$  will be denoted by  $q_i$ .
- While carrying out step  $i$  of Euclidean algorithm, also calculate auxiliary number,  $p_i$ .
- For first two steps, value of this number is given by:  
 $p_0 = 0$  and  $p_1 = 1$ .
- For remaining steps, recursively calculate  
 $p_i = p_{i-2} - p_{i-1}q_{i-2} \pmod{n}$ .
- Continue calculation for one step beyond last step of Euclidean algorithm.

- The algorithm starts by “dividing”  $n$  by  $x$ .
- If last non-zero remainder occurs at step  $k$ , then  
if this remainder is 1,  $x$  has an inverse and it is  $p_{k+2}$ .  
(If remainder is not 1, then  $x$  does **not** have an inverse.)

### Example:

Find the inverse of 15 mod 26.

$$p_i = p_{i-2} - p_{i-1}q_{i-2} \pmod{n}$$

Step 0:  $26 = 1(15) + 11$

$$p_0 = 0$$

Step 1:  $15 = 1(11) + 4$

$$p_1 = 1$$

Step 2:  $11 = 2(4) + 3$

$$p_2 = 0 - 1(1) \pmod{26} = 25$$

Step 3:  $4 = 1(3) + 1$

$$p_3 = 1 - 25(1) \pmod{26} = 2$$

Step 4:  $3 = 3(1) + 0$

$$p_4 = 25 - 2(2) \pmod{26} = 21$$

$$p_5 = 2 - 21(1) \pmod{26} = 7$$