

CSCI 124/224

Discrete Structures II: Generalized Euclidean Algorithm to Determine Inverse in \mathbb{Z}_m

Poorvi L. Vora

We have observed that the euclidean algorithm can be used to find the GCD of two integers, and hence can be used to determine if the integer a has a multiplicative inverse in \mathbb{Z}_m . In this section, we see how we may determine the value of the inverse if a is invertible.

Recall the euclidean algorithm. For example, recall its use to determine $gcd(79, 551)$.

$$\begin{aligned}
 (a, b) &= (551, 79) \\
 (a, b) &= (79, 77) \\
 (a, b) &= (77, 2) \\
 (a, b) &= (2, 1) \\
 (a, b) &= (1, 0) \\
 &return(1)
 \end{aligned}$$

As the gcd of 79 and 551 is 1, 79 is invertible *modulo* 551. In fact, the euclidean algorithm can be reversed as follows to determine the inverse, by keeping a record of the quotient.

| i | a | b | q_i |
|-----|-----|-----|-------|
| 0 | 551 | 79 | 6 |
| 1 | 79 | 77 | 1 |
| 2 | 77 | 2 | 38 |
| 3 | 2 | 1 | 2 |
| 4 | 1 | 0 | |

Now reverse direction and compute an extra pair, $(s, t) := (t, s - t * q_i)$.

| i | a | b | q_i | s | t |
|-----|-----|-----|-------|-----|--------------------|
| 0 | 551 | 79 | 6 | 39 | $-38-(39)6 = -272$ |
| 1 | 79 | 77 | 1 | -38 | $1 - (-38)(1)=39$ |
| 2 | 77 | 2 | 38 | 1 | $0-(1)(38)=-38$ |
| 3 | 2 | 1 | 2 | 0 | 1 |
| 4 | 1 | 0 | | | |

What you've found is the numbers s and t such that $sa + tb = 1$. Thus $39 \times 551 + (-272) \times 79 = 1$, and $(-272) \times 79 = 1 \pmod{551}$, or $279 \times 79 = 1 \pmod{551}$, and $79^{-1} \pmod{551} = 279$.

```

gcd_and_inverse(m, n) /* m > n */
(a, b) := (m, n) /* Initialize */
i = 0; /* Keep track of all quotients */
while (b ≠ 0)
{
  qi := ⌊ $\frac{a}{b}$ ⌋ /* Remember  $i^{th}$  quotient */
  (a, b) := (b, a mod b) /* Reduce problem */
  i := i+1 /* Increment count */
}
i := i-1; /* Now i reflects the number of q's because the last value of i does not correspond to a q */
/* Now go back up loop to determine inverse */
(s, t) := (0, 1) /* Initialize */
i := i-1 /* Update i */
while (i > 0)
{
  (s, t) := (t, s - t * qi)
  i := i-1
}
return(a, (s,t))

```

Example: Use the generalized euclidean algorithm to determine the decryption key for an affine cipher if the encryption key is (28, 7) and the modulus is 75; that is, the encryption function is $e_K(x) = 28x + 7 \pmod{75}$.

First find $28^{-1} \pmod{75}$.

| i | a | b | q_i |
|-----|-----|-----|-------|
| 0 | 75 | 28 | 2 |
| 1 | 28 | 19 | 1 |
| 2 | 19 | 9 | 2 |
| 3 | 9 | 1 | 9 |
| 4 | 1 | 0 | |

Now reverse direction.

| i | a | b | q_i | s | t |
|-----|-----|-----|-------|-----|-------------------|
| 0 | 75 | 28 | 2 | 3 | $-2-(3)(2) = -8$ |
| 1 | 28 | 19 | 1 | -2 | $1 - (-2)(1) = 3$ |
| 2 | 19 | 9 | 2 | 1 | $0-(1)(2) = -2$ |
| 3 | 9 | 1 | 9 | 0 | 1 |
| 4 | 1 | 0 | | | |

Thus $3 \times 75 + (-8) \times 28 = 1$, and $(-8) \times 28 = 1 \pmod{75}$, or $67 \times 28 = 1 \pmod{75}$, and $28^{-1} \pmod{75} = 67$.

$$\begin{aligned}y = e_K(x) &= 28x + 7 \pmod{75} \\ \Rightarrow x &= 28^{-1}(y - 7) \pmod{75} \\ \Rightarrow x &= 67(y - 7) \pmod{75} \\ \Rightarrow x &= 67y + 56 \pmod{75} \\ \Rightarrow d_K(x) &= 67x + 56 \pmod{75}\end{aligned}$$

and the decryption key is $(67, 56)$.