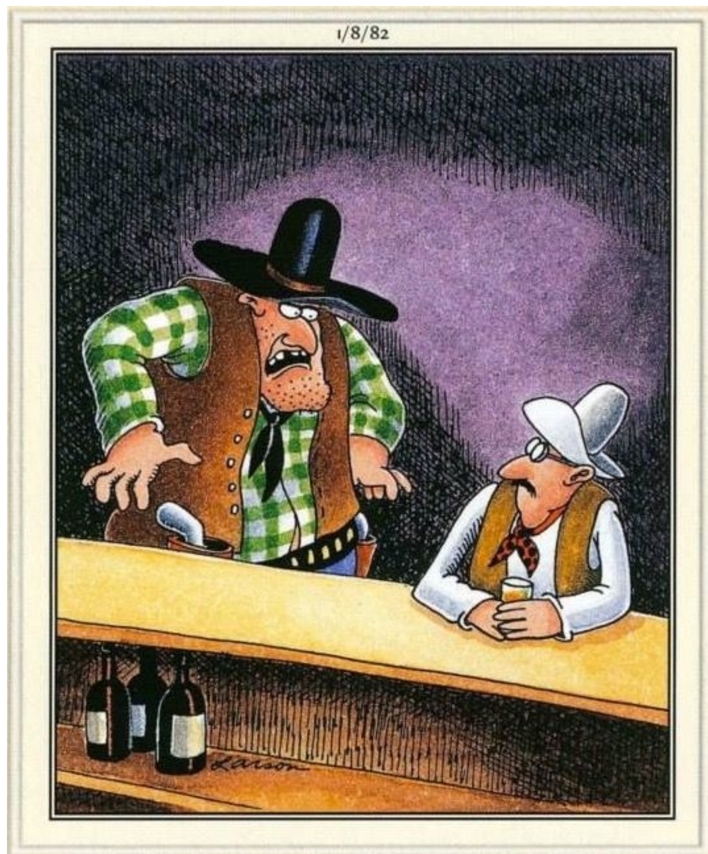
CSE 311: Foundations of Computing

Lecture 14: Modular Inverse, Exponentiation



"I *asked* you a question, buddy. ... What's the square root of 5,248?"

Last time: Useful GCD Facts

If a and b are positive integers, then

$$\gcd(a, b) = \gcd(b, a \bmod b)$$

If a is a positive integer, $\gcd(a, 0) = a$.

Euclid's Algorithm

gcd(a, b) = gcd(b, a mod b)   gcd(a, 0) = a

```
int gcd(int a, int b){ /* Assumes: a >= b, b >= 0 */
    if (b == 0) {
        return a;
    } else {
        return gcd(b, a % b);
    }
}
```

Note: gcd(b, a) = gcd(a, b)

Euclid's Algorithm

Repeatedly use $\gcd(a, b) = \gcd(b, a \bmod b)$ to reduce numbers until you get $\gcd(g, 0) = g$.

gcd(660,126) =

Euclid's Algorithm

Repeatedly use $\gcd(a, b) = \gcd(b, a \bmod b)$ to reduce numbers until you get $\gcd(g, 0) = g$.

gcd(660,126) = gcd(126, 660 mod 126) = gcd(126, 30)
= gcd(30, 126 mod 30) = gcd(30, 6)
= gcd(6, 30 mod 6) = gcd(6, 0)
= 6

Euclid's Algorithm

Repeatedly use $\gcd(a, b) = \gcd(b, a \bmod b)$ to reduce numbers until you get $\gcd(g, 0) = g$.

Equations with recursive calls:

gcd(660,126) = gcd(126, 660 mod 126) = gcd(126, 30)
= gcd(30, 126 mod 30) = gcd(30, 6)
= gcd(6, 30 mod 6) = gcd(6, 0)
= 6

Tableau form:

660 = 5 * 126 + 30
126 = 4 * 30 + 6
30 = 5 * 6 + 0

Bézout's theorem

If a and b are positive integers, then there exist integers $\boldsymbol{s}$ and $\boldsymbol{t}$ such that

$$\gcd(a,b) = \boldsymbol{s}a + \boldsymbol{t}b.$$

$$\forall a\, \forall b\, ((a > 0 \land b > 0) \rightarrow \exists s\, \exists t\, (\gcd(a,b) = sa + tb))$$

Extended Euclidean algorithm

- Can use Euclid's Algorithm to find s, t such that

$$\gcd(a, b) = sa + tb$$

Extended Euclidean algorithm

- Can use Euclid's Algorithm to find s, t such that

$$\gcd(a, b) = sa + tb$$

Step 1 (Compute GCD & Keep Tableau Information):

$$\gcd(\underset{a}{35}, \underset{b}{27}) = \gcd(\underset{b}{27}, \underbrace{\underset{a}{35} \bmod \underset{b}{27}}_{= r}) = \gcd(\underset{b}{27}, \underset{r}{8})$$

$$\underset{a}{35} = \underset{q}{1} * \underset{b}{27} + \underset{r}{8}$$

Extended Euclidean algorithm

- Can use Euclid's Algorithm to find s, t such that

$$\gcd(a, b) = sa + tb$$

Step 1 (Compute GCD & Keep Tableau Information):

$$\begin{aligned}
&\ \ \ \overset{a}{\ }\ \ \overset{b}{\ } && \ \ \ \text{b}\ \ \ \text{a mod b = r} && \ \ \ \text{b}\ \ \ \text{r} \\
\gcd(35, 27) &= \gcd(27, 35 \bmod 27) &&= \gcd(27, 8) \\
&= \gcd(8, 27 \bmod 8) &&= \gcd(8, 3) \\
&= \gcd(3, 8 \bmod 3) &&= \gcd(3, 2) \\
&= \gcd(2, 3 \bmod 2) &&= \gcd(2, 1) \\
&= \gcd(1, 2 \bmod 1) &&= \gcd(1, 0)
\end{aligned}$$

$$\boxed{\begin{aligned}
a &= q * b + r \\
35 &= 1 * 27 + 8 \\
27 &= 3 * 8 + 3 \\
8 &= 2 * 3 + 2 \\
3 &= 1 * 2 + 1
\end{aligned}}$$

Extended Euclidean algorithm

- Can use Euclid's Algorithm to find s, t such that

$$\gcd(a, b) = sa + tb$$

Step 2 (Solve the equations for r):

$$\boxed{\begin{array}{l} a = q * b + r \\ 35 = 1 * 27 + 8 \\ 27 = 3 * 8 + 3 \\ 8 = 2 * 3 + 2 \\ 3 = 1 * 2 + 1 \end{array}}$$

r = a – q * b

$8 = 35 – 1 * 27$

Extended Euclidean algorithm

- **Can use Euclid's Algorithm to find s, t such that**

$$\gcd(a, b) = sa + tb$$

Step 2 (Solve the equations for r):

a = q * b + r	r = a – q * b
$35 = 1 * 27 + 8$	$8 = 35 - 1 * 27$
$27 = 3 * 8 + 3$	$3 = 27 - 3 * 8$
$8 = 2 * 3 + 2$	$2 = 8 - 2 * 3$
$3 = 1 * 2 + 1$	$1 = 3 - 1 * 2$

Extended Euclidean algorithm

- Can use Euclid's Algorithm to find s, t such that

$$\gcd(a, b) = sa + tb$$

Step 3 (Backward Substitute Equations):

$$8 = 35 - 1 * 27$$

$$3 = 27 - 3 * 8$$

$$2 = 8 - 2 * 3$$

$$\boxed{1 = 3 - 1 * 2}$$

Extended Euclidean algorithm

- Can use Euclid's Algorithm to find s, t such that

$$\gcd(a, b) = sa + tb$$

Step 3 (Backward Substitute Equations):

Plug in the def of 2

$$8 = 35 - 1 * 27$$

$$3 = 27 - 3 * 8$$

$$2 = 8 - 2 * 3$$

$$\boxed{1 = 3 - 1 * 2}$$

$$\begin{aligned} 1 &= 3 - 1 * (8 - 2 * 3) \\ &= 3 - 8 + 2 * 3 \\ &= (-1) * 8 + 3 * 3 \end{aligned}$$

Re-arrange into
3's and 8's

Extended Euclidean algorithm

- Can use Euclid's Algorithm to find s, t such that

$$\gcd(a, b) = sa + tb$$

Step 3 (Backward Substitute Equations):

$$8 = 35 - 1 * 27$$

$$3 = 27 - 3 * 8$$

$$2 = 8 - 2 * 3$$

$$\boxed{1 = 3 - 1 * 2}$$

Plug in the def of 2

$$\begin{aligned} 1 &= 3 - 1 * (8 - 2 * 3) \\ &= 3 - 8 + 2 * 3 \\ &= (-1) * 8 + 3 * 3 \end{aligned}$$

Re-arrange into 3's and 8's

Plug in the def of 3

$$\begin{aligned} &= (-1) * 8 + 3 * (27 - 3 * 8) \\ &= (-1) * 8 + 3 * 27 + (-9) * 8 \\ &= 3 * 27 + (-10) * 8 \end{aligned}$$

Re-arrange into 8's and 27's

Extended Euclidean algorithm

- Can use Euclid's Algorithm to find s, t such that

$$\gcd(a, b) = sa + tb$$

Step 3 (Backward Substitute Equations):

$8 = 35 - 1 * 27$

$3 = 27 - 3 * 8$

$2 = 8 - 2 * 3$

$\boxed{1 = 3 - 1 * 2}$

Plug in the def of 2

$$\begin{aligned} 1 &= 3 - 1 * (8 - 2 * 3) \\ &= 3 - 8 + 2 * 3 \\ &= (-1) * 8 + 3 * 3 \end{aligned}$$

Re-arrange into 3's and 8's

Plug in the def of 3

$$\begin{aligned} &= (-1) * 8 + 3 * (27 - 3 * 8) \\ &= (-1) * 8 + 3 * 27 + (-9) * 8 \\ &= 3 * 27 + (-10) * 8 \end{aligned}$$

Re-arrange into 8's and 27's

$$\begin{aligned} &= 3 * 27 + (-10) * (35 - 1 * 27) \\ &= 3 * 27 + (-10) * 35 + 10 * 27 \\ &= 13 * 27 + (-10) * 35 \end{aligned}$$

Re-arrange into 27's and 35's

Multiplicative inverse $\mathrm{mod}\ m$

Let $0 \leq a, b < m$. Then, b is the *multiplicative inverse of* a *(modulo* m*)* iff $ab \equiv_m 1$.

x	0	**1**	**2**	**3**	**4**	**5**	**6**
0	0	0	0	0	0	0	0
1	0	**1**	2	3	4	5	6
2	0	2	4	6	**1**	3	5
3	0	3	6	2	5	**1**	4
4	0	4	**1**	5	2	6	3
5	0	5	3	**1**	6	4	2
6	0	6	5	4	3	2	**1**

mod 7

x	0	**1**	2	**3**	4	5	6	**7**	8	**9**
0	0	0	0	0	0	0	0	0	0	0
1	0	**1**	2	3	4	5	6	7	8	9
2	0	2	4	6	8	0	2	4	6	8
3	0	3	6	9	2	5	8	**1**	4	7
4	0	4	8	2	6	0	4	8	2	6
5	0	5	0	5	0	5	0	5	0	5
6	0	6	2	8	4	0	6	2	8	4
7	0	7	4	**1**	8	5	2	9	6	3
8	0	8	6	4	2	0	8	6	4	2
9	0	9	8	7	6	5	4	3	2	**1**

mod 10

Multiplicative inverse $\mathrm{mod}\ m$

Suppose $\gcd(a, m) = 1$

By Bézout's Theorem, there exist integers s and t such that $sa + tm = 1$.

s is the multiplicative inverse of a (modulo m):

$$1 = sa + tm \equiv_m sa$$

So... we can compute multiplicative inverses with the extended Euclidean algorithm

These inverses let us solve modular equations...

Example

Solve: $7x \equiv_{26} 1$

Example

Solve: $7x \equiv_{26} 1$

$$\gcd(26,7) = \gcd(7,5) = \gcd(5,2) = \gcd(2,1) = 1$$

Example

Solve: $7x \equiv_{26} 1$

$$\gcd(26,7) = \gcd(7,5) = \gcd(5,2) = \gcd(2,1) = 1$$

$$26 = 3 * 7 + 5$$
$$7 = 1 * 5 + 2$$
$$5 = 2 * 2 + 1$$

Example

Solve: $7x \equiv_{26} 1$

$$\gcd(26,7) = \gcd(7,5) = \gcd(5,2) = \gcd(2,1) = 1$$

$$\begin{aligned} 26 &= 3 * 7 + 5 \qquad & 5 &= 26 - 3 * 7 \\ 7 &= 1 * 5 + 2 \qquad & 2 &= 7 - 1 * 5 \\ 5 &= 2 * 2 + 1 \qquad & 1 &= 5 - 2 * 2 \end{aligned}$$

Example

Solve: $7x \equiv_{26} 1$

$$\gcd(26,7) = \gcd(7,5) = \gcd(5,2) = \gcd(2,1) = 1$$

$$\begin{aligned} 26 &= 3 * 7 + 5 & \qquad 5 &= 26 - 3 * 7 \\ 7 &= 1 * 5 + 2 & \qquad 2 &= 7 - 1 * 5 \\ 5 &= 2 * 2 + 1 & \qquad 1 &= 5 - 2 * 2 \end{aligned}$$

$$\begin{aligned} 1 &= 5 - 2 * (7 - 1 * 5) \\ &= (-2) * 7 + 3 * 5 \\ &= (-2) * 7 + 3 * (26 - 3 * 7) \\ &= (-11) * 7 + 3 * 26 \end{aligned}$$

Example

Solve: $7x \equiv_{26} 1$

$$\gcd(26,7) = \gcd(7,5) = \gcd(5,2) = \gcd(2,1) = 1$$

$$\begin{aligned} 26 &= 3*7 + 5 & \qquad 5 &= 26 - 3*7 \\ 7 &= 1*5 + 2 & 2 &= 7 - 1*5 \\ 5 &= 2*2 + 1 & 1 &= 5 - 2*2 \end{aligned}$$

$$\begin{aligned} 1 &= 5 - 2*(7 - 1*5) \\ &= (-2)*7 + 3*5 \\ &= (-2)*7 + 3*(26 - 3*7) \\ &= (-11)*7 + 3*26 \end{aligned}$$

Multiplicative inverse of 7 modulo 26

Now $(-11) \bmod 26 = 15$. **So,** $x = 15 + 26k$ **for** $k \in \mathbb{Z}$.

Example of a more general equation

Now solve: $7y \equiv_{26} 3$

We already computed that 15 is the multiplicative inverse of 7 modulo 26. That is, $7 \cdot 15 \equiv_{26} 1$

If y is a solution, then multiplying by 15 we have

$$15 \cdot 7 \cdot y \equiv_{26} 15 \cdot 3$$

Substituting $15 \cdot 7 \equiv_{26} 1$ into this on the left gives

$$\mathrm{y} = 1 \cdot y \equiv_{26} 15 \cdot 3 \equiv_{26} 19$$

This shows that <u>every</u> solution y is congruent to 19.

Example of a more general equation

Now solve: $7y \equiv_{26} 3$

Multiplying both sides of $y \equiv_{26} 19$ by 7 gives

$$7\mathrm{y} \equiv_{26} 7 \cdot 19 \equiv_{26} 3$$

So, <u>any</u> $y \equiv_{26} 19$ is a solution.

Thus, the set of numbers of the form $y = 19 + 26k$, for any k, are <u>exactly</u> solutions of this equation.

Math mod a prime is especially nice

$\gcd(a, m) = 1$ **if** m **is prime and** $0 < a < m$ **so can always solve these equations mod a prime.**

+	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

x	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

mod 7

Multiplicative Inverses and Algebra

Adding to both sides is an equivalence:

$$\begin{tikzcd} x \equiv_m y \arrow[d, bend left, "+c"] \\ x + c \equiv_m y + c \arrow[u, bend left, "-c"] \end{tikzcd}$$

The same is not true of multiplication...

unless we have a multiplicative inverse $cd \equiv_m 1$

$$\begin{tikzcd} x \equiv_m y \arrow[d, bend left, "\times c"] \\ cx \equiv_m cy \arrow[u, bend left, "\times d"] \end{tikzcd}$$

Modular Exponentiation mod 7

x	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	1	3	5
3	3	6	2	5	1	4
4	4	1	5	2	6	3
5	5	3	1	6	4	2
6	6	5	4	3	2	1

a	a^1	a^2	a^3	a^4	a^5	a^6
1	1	1	1	1	1	1
2	2	4	1	2	4	1
3	3	2	6	4	5	1
4	4	2	1	4	2	1
5	5	4	6	2	3	1
6	6	1	6	1	6	1

Exponentiation

- **Compute** 78365^{81453}

- **Compute** $78365^{81453} \bmod 104729$

- **Output is small**
  - need to keep intermediate results small

Small Multiplications

Since $b = qm + (b \bmod m)$, we have $b \bmod m \equiv_m b$.

And since $c = tm + (c \bmod m)$, we have $c \bmod m \equiv_m c$.

Multiplying these gives $(b \bmod m)(c \bmod m) \equiv_m bc$.

By the Lemma from a few lectures ago, this tells us $bc \bmod m = (b \bmod m)(c \bmod m) \bmod m$.

Okay to mod b and c by m before multiplying if we are planning to mod the result by m

Repeated Squaring – small and fast

Since $b \bmod m \equiv_m b$ and $c \bmod m \equiv_m c$

we have $bc \bmod m = (b \bmod m)(c \bmod m) \bmod m$

So $\quad a^2 \bmod m = (a \bmod m)^2 \bmod m$

and $\quad a^4 \bmod m = (a^2 \bmod m)^2 \bmod m$

and $\quad a^8 \bmod m = (a^4 \bmod m)^2 \bmod m$

and $\quad a^{16} \bmod m = (a^8 \bmod m)^2 \bmod m$

and $\quad a^{32} \bmod m = (a^{16} \bmod m)^2 \bmod m$

Can compute $a^k \bmod m$ for $k = 2^i$ in only i steps

What if k is not a power of 2?

Fast Exponentiation Algorithm

81453 in binary is 10011111000101101

$81453 = 2^{16} + 2^{13} + 2^{12} + 2^{11} + 2^{10} + 2^{9} + 2^{5} + 2^{3} + 2^{2} + 2^{0}$

$a^{81453} = a^{2^{16}} \cdot a^{2^{13}} \cdot a^{2^{12}} \cdot a^{2^{11}} \cdot a^{2^{10}} \cdot a^{2^{9}} \cdot a^{2^{5}} \cdot a^{2^{3}} \cdot a^{2^{2}} \cdot a^{2^{0}}$

$$
\begin{aligned}
a^{81453} \bmod m = \\
(\ldots(((((a^{2^{16}} \bmod m \cdot \\
&a^{2^{13}} \bmod m\ ) \bmod m \cdot \\
&\quad a^{2^{12}} \bmod m) \bmod m \cdot \\
&\qquad a^{2^{11}} \bmod m) \bmod m \cdot \\
&\qquad\quad a^{2^{10}} \bmod m) \bmod m \cdot \\
&\qquad\qquad a^{2^{9}} \bmod m) \bmod m \cdot \\
&\qquad\qquad\quad a^{2^{5}} \bmod m) \bmod m \cdot \\
&\qquad\qquad\qquad a^{2^{3}} \bmod m) \bmod m \cdot \\
&\qquad\qquad\qquad\quad a^{2^{2}} \bmod m) \bmod m \cdot \\
&\qquad\qquad\qquad\qquad a^{2^{0}} \bmod m) \ \bmod m
\end{aligned}
$$

Uses only 16 + 9 = 25 multiplications

The fast exponentiation algorithm computes $a^k \bmod m$ using $\leq 2\log k$ multiplications $\bmod m$

Fast Exponentiation: $a^k \bmod m$ for all k

Another way....

$$a^{2j} \bmod m = \left(a^j \bmod m\right)^2 \bmod m$$

$$a^{2j+1} \bmod m = \left((a \bmod m) \cdot (a^{2j} \bmod m)\right) \bmod m$$

Fast Exponentiation

```
public static int FastModExp(int a, int k, int modulus) {

        if (k == 0) {
            return 1;

        } else if ((k % 2) == 0) {
            long temp = FastModExp(a,k/2,modulus);
            return (temp * temp) % modulus;

        } else {
            long temp = FastModExp(a,k-1,modulus);
            return (a * temp) % modulus;
        }
}
```

$$a^{2j} \bmod m = \left(a^{j} \bmod m\right)^{2} \bmod m$$

$$a^{2j+1} \bmod m \ = \ \left((a \bmod m) \cdot (a^{2j} \bmod m)\right) \bmod m$$

Using Fast Modular Exponentiation

- Your e-commerce web transactions use SSL (Secure Socket Layer) based on RSA encryption
- RSA
  - Vendor chooses random 512-bit or 1024-bit primes p, q and 512/1024-bit exponent e. Computes $m = p \cdot q$
  - Vendor broadcasts (m, e)
  - To send a to vendor, you compute $C = a^e \bmod m$ using *fast modular exponentiation* and send C to the vendor.
  - Using secret p, q the vendor computes d that is the *multiplicative inverse* of e mod $(p-1)(q-1)$.
  - Vendor computes $C^d \bmod m$ using *fast modular exponentiation*.
  - **Fact:** $a = C^d \bmod m$ for $0 < a < m$ unless $p|a$ or $q|a$