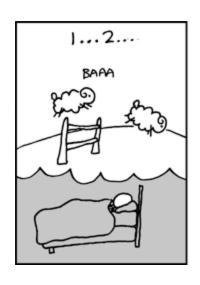
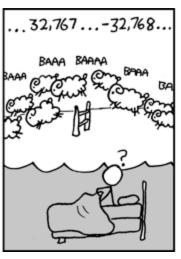
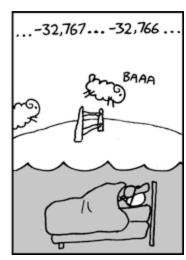
CSE 311: Foundations of Computing

Lecture 15: The Euclidean algorithm and applications









Recap from last lecture

- An integer p>2 is prime if the only positive factors are 1 and p
- A prime factorization for an integer n>1 is of the form $n=p_1\cdot\ldots\cdot p_k$ where p_1,\ldots,p_k are prime numbers.
- Each integer n > 1 as a unique prime factorization.
- The greatest common divisor gcd(a, b) is the largest integer d with $d \mid a$ and $d \mid b$.
- Important fact for today: gcd(a, b) = gcd(b, a % b)

Another simple GCD fact

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

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

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

Example: GCD(660, 126)

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

gcd(660,126) =

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

```
gcd(660,126) = gcd(126, 660 % 126) = gcd(126, 30)
= gcd(30, 126 % 30) = gcd(30, 6)
= gcd(6, 30 % 6) = gcd(6, 0)
= 6
```

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

In tableau form:
$$9 \cdot 3 + 6$$

 $660 = 5 * 126 + 30$
 $126 = 4 * 30 + 6$
 $30 = 5 * 6 + 0$
In tableau form: $30 = 3 \cdot 30 = 30$
 $30 = 5 \cdot 6 + 0$
 $30 = 5 \cdot 6 + 0$

Bézout's theorem

If a and b are positive integers, then there exist integers s and t such that gcd(a,b) = sa + tb.

Proof is algorithmic: via the Extended Euclidean algorithm

- Method: $extGCD(a, b) \rightarrow (g, s, t)$
- Input: Integers $a \ge b \ge 0$
- Output: Integers g, s, t s.t. $g = \gcd(a, b) = s \cdot a + t \cdot b$
- **1.** If b == 0 THEN return (a, 1, 0) // $a = \gcd(a, 0) = 1 \cdot a + 0 \cdot 0$
- 2. (g, s, t) = extGCD(b, a % b)
- 3. Write a = qb + (a % b) with $q \in \mathbb{Z}$
- 4. Return (g, t, s tq)

```
//g = \gcd(b, a \% b) = \gcd(a, b)
//g = s \cdot b + t \cdot (a \% b)
// = s \cdot b + t \cdot (a - qb)
// = t \cdot a + (s - tq) \cdot b
```

• Example: Find s, t such that $gcd(31,16) = s \cdot 31 + t \cdot 16$

extGCD(31, 16) -> extGCD(16, 31 % 16)

```
extGCD(31, 16) -> extGCD(16, 31 % 16)
extGCD(16, 15) -> extGCD(15, 16 % 15)
```

```
extGCD(31, 16) -> extGCD(16, 31 % 16)
extGCD(16, 15) -> extGCD(15, 16 % 15)
extGCD(15, 1) -> extGCD(1, 15 % 1)
```

```
extGCD(31, 16) -> extGCD(16, 31 % 16)

extGCD(16, 15) -> extGCD(15, 16 % 15)

extGCD(15, 1) -> extGCD(1, 15 % 1)

extGCD(1, 0) = 1 = 1*1 + 0*0
```

```
extGCD(31, 16) -> extGCD(16, 31 % 16)

extGCD(16, 15) -> extGCD(15, 16 % 15)

extGCD(15, 1) -> extGCD(1, 15 % 1)

extGCD(1, 0) = 1 = 1*1 + 0*0

15=15*1 + 0

1 = 0*15 + (1 - 0*15)*1 = 0*15 + 1*1
```

Hence
$$gcd(31,16) = 1$$

Hence
$$gcd(31,16) = 1$$

Hence
$$gcd(31,16) = 1$$

• Example: Find s, t such that $gcd(31,16) = s \cdot 31 + t \cdot 16$

```
extGCD(31, 16) -> extGCD(16, 31 % 16)
       extGCD(16, 15) -> extGCD(15, 16 % 15)
        extGCD(15, 1) -> extGCD(1, 15 % 1)
          extGCD(1, 0) = 1 = 1*1 + 0*0
        15=15<u>*1</u> + 0
        1 = 0*15 + (1-0*15)*1 = 0*15 + 1*1
       16=1*15 + 1
31=1*16+(0-1*1)*15=1*16+-1*15
31=1*16+15
Stept 1 = -1*31 + (1 - -1*1)*16 = -1*31 + 2*16
```

$$1 = 1.6 \times -1315$$

$$1 = 1.6 \times -1316$$

$$1 = 1.16 \times (-1)(31-1.6)$$

Hence $gcd(31,16) = 1 = (-1) \cdot 31 + 2 \cdot 16$

```
extGCD(31, 16) -> extGCD(16, 31 % 16)
 extGCD(16, 15) -> extGCD(15, 16 % 15)
   extGCD(15, 1) -> extGCD(1, 15 % 1)
    extGCD(1, 0) = 1 = 1*1 + 0*0
   15=15*1 + 0
   1 = 0*15 + (1-0*15)*1 = 0*15 + 1*1
 16=1*15+1
 1 = 1*16 + (0 - 1*1)*15 = 1*16 + -1*15
31=1*16 + 15
1 = -1*31 + (1 - -1*1)*16 = -1*31 + 2*16
```

Hence
$$gcd(31,16) = 1 = (-1) \cdot 31 + 2 \cdot 16$$

```
extGCD(31, 16) -> extGCD(16, 31 % 16)
 extGCD(16, 15) -> extGCD(15, 16 % 15)
   extGCD(15, 1) -> extGCD(1, 15 % 1)
    extGCD(1, 0) = 1 = 1*1 + 0*0
   15=15*1+0
   1 = 0*15 + (1 - 0*15)*1 = 0*15 + 1*1
 16=1*15+1
 1 = 1*16 + (0 - 1*1)*15 = 1*16 + -1*15
31=1*16+15
1 = -1*31 + (1 - -1*1)*16 = -1*31 + 2*16
```

Hence
$$gcd(31,16) = 1 = (-1) \cdot 31 + 2 \cdot 16$$

Multiplicative inverse $\mod m$

Suppose
$$GCD(a, m) = 1$$

By Bézout's Theorem, there exist integers s and t such that sa + tm = 1. 12m=5a2m 1=5a (modm) s % m is the multiplicative inverse of $a: sa = 1 \pmod{m}$ $1 = (sa + tm) \frac{\% m}{5m} = sa \% m \qquad sa \cdot 5 \equiv 5 \pmod{m}$ suppose a.x=1 hulm $sa = 1 \pmod{m}$ then 32: X=8 [mdn]
1. X=5 [mdn] 2x=Zy < x=y (mdm)

Suppose
$$GCD(a, m) = 1$$

$$5 ce = 1$$
 $5 cx = 5$
 $5 cx = 5$

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

s % m is the multiplicative inverse of $a: sa = 1 \pmod{m}$

$$1 = (sa + tm) \% m = sa \% m$$

$$Sa=1$$
 (mod m) Suppose $a.x=1$ ($x=8$ 3)

 $2x=2y \leftarrow x=y \pmod{n}$ then $3x:x=8$ 3)

 $x\cdot 2+x+z\cdot x=y\cdot 2+y+2y \pmod{n}$

Example

Solve:
$$7x \equiv 1 \pmod{26}$$

$$57x \equiv 1 \pmod{26}$$

Example

58x = 5 (mod 26)

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

$$26 = 7 * 3 + 5$$

$$7 = 5 * 1 + 2$$

$$5 = 2 * 2 + 1$$

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

$$5 = 26 - 7 * 3$$

$$2 = 7 - 5 * 1$$

$$1 = 5 - 2 * 2$$

$$1 = 5 - 2*(7-5*1) \\
= (-7)*2 + 3*5 \\
= (-7)*2 + 3*(26-7*3)$$

$$7.(-11) = -77$$

$$7.(5 = 105$$

$$= 4.26 + 1$$

= (-11) * 7 + 3 * 26

Multiplicative inverse of 7 mod 26

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

Example of a more general equation

Now solve:
$$7y \equiv 3 \pmod{26}$$
 s \neq

We already computed that 15 is the multiplicative inverse of 7 modulo 26:

That is,
$$7 \cdot 15 \equiv 1 \pmod{26}$$

By the multiplicative property of mod we have

$$7 \cdot 15 \cdot 3 \equiv 3 \pmod{26}$$

So any $y \equiv 15 \cdot 3 \pmod{26}$ is a solution.

That is, y = 19 + 26k for any integer k is a solution.

Lecture 15 Activity

You will be assigned to breakout rooms. Please:

- Introduce yourself
- Choose someone to share their screen, showing this PDF
- Discuss the following questions:
- If you run the extended Euclidean algorithm for (51,23) it will return that $gcd(51,23) = 1 = (-9) \cdot 51 + 20 \cdot 23$. What is this telling you about the multiplicative inverse of 51 modulo 23.
- 2. If you run the extended Euclidean algorithm for (51,24) it will return that $gcd(51,24) = 3 = 1 \cdot 51 + (-2) \cdot 24$. What is this telling you about the multiplicative inverse of 51 modulo 24.
- 3. What is the set of integers that do not have a multiplicative inverse modulo 10?

Fill out the poll everywhere for Activity Credit!
Go to pollev.com/philipmg and login with your UW identity

Lecture 15 Activity

1. If you run the extended Euclidean algorithm for (51,23) it will return that $gcd(51,23) = 1 = (-9) \cdot 51 + 20 \cdot 23$. What is this telling you about the multiplicative inverse of 51 modulo 23.

Solution: -9 + 23 = 14

2. If you run the extended Euclidean algorithm for (51,24) it will return that $gcd(51,24) = 3 = 1 \cdot 51 + (-2) \cdot 24$. What is this telling you about the multiplicative inverse of 51 modulo 24.

Solution: There is none.

3. What is the set of integers that do not have a multiplicative inverse modulo 10?

Solution: 0 + 10x, 2 + 20x, 5 + 10x, $x \in \mathbb{Z}$

Fill out the poll everywhere for Activity Credit!

Go to pollev.com/philipmg and login with your UW identity

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

Х	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

Modular Exponentiation % 7

Х	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

а	a ¹	a ²	a ³	a ⁴	a ⁵	a ⁶
1						
2						
3						
4						
5						
6						

Exponentiation

• Compute 78365⁸¹⁴⁵³

• Compute 78365⁸¹⁴⁵³ % 104729

- Output is small
 - need to keep intermediate results small

Repeated Squaring – small and fast

```
Since a \% m \equiv a \pmod{m} and b \% m \equiv b \pmod{m}
we have ab \% m = ((a \% m)(b \% m))\% m
         a^2 \% m = (a \% m)^2 \% m
So
         a^4 \% m = (a^2 \% m)^2 \% m
and
         a^8 \% m = (a^4 \% m)^2 \% m
and
         a^{16} \% m = (a^8 \% m)^2 \% m
and
         a^{32} \% m = (a^{16} \% m)^2 \% m
and
```

Can compute $a^k \% m$ 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}
 a<sup>81453</sup> % m=
a^{2^{13}} /% m · a^{2^{13}} /% m · a^{2^{13}} /% m ) % m · a^{2^{13}} /% m ) % m · a^{2^{13}}
                     a<sup>212</sup> % m) % m -
                         a<sup>2<sup>11</sup></sup> % m) % m - a<sup>2<sup>10</sup></sup> % m) % m -
                                  a<sup>29</sup> % m) % m -
                                       a<sup>25</sup> % m) % m ·
                                             a<sup>23</sup> % m) % m -
                                                    a<sup>2<sup>2</sup></sup> % m) % m ·
                                                         a<sup>20</sup> % m) % m
```

The fast exponentiation algorithm computes

 a^k % m using $\leq 2\log k$ multiplications % m

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

Another way....

$$a^{2j}\% m = (a^j \% m)^2\% m$$

 $a^{2j+1}\% m = ((a \% m) \cdot (a^{2j} \% m)) \% m$

Fast Exponentiation

```
public static long FastModExp(long a, long k, long modulus) {
       long result = 1;
       long temp;
       if (k > 0) {
           if ((k \% 2) == 0) {
               temp = FastModExp(a,k/2,modulus);
               result = (temp * temp) % modulus;
           }
           else {
               temp = FastModExp(a,k-1,modulus);
               result = (a * temp) % modulus;
      return result;
    a^{2j}\% m = (a^{j}\% m)^{2}\% m
    a^{2j+1}\% m = ((a \% m) \cdot (a^{2j} \% m)) \% 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 $C = a^e \% m$ 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 $C^d \% m$ using fast modular exponentiation.
- Fact: $a = C^d \% m$ for 0 < a < m unless p|a or q|a