CSE 311 Foundations of Computing I
Lecture 12
Primes, GCD, Modular Inverse
Spring 2013

Announcements

• Reading assignments
  – Today:
    • 7th Edition: 4.3-4.4 (the rest of the chapter is interesting!)
    • 6th Edition: 3.5, 3.6
  – Monday: Mathematical Induction
    • 7th Edition: 5.1, 5.2
    • 6th Edition: 4.1, 4.2

Fast modular exponentiation

Fast exponentiation algorithm

• What if the exponent is not a power of two?

\[ 81453 = 2^{16} + 2^{13} + 2^{12} + 2^{11} + 2^{10} + 2^9 + 2^8 + 2^3 + 2^2 + 2^0 \]

\[ 78365^{81453} = 78365^{2^{16}} 78365^{2^{13}} 78365^{2^{12}} 78365^{2^{11}} \ldots \]

The fast exponentiation algorithm computes \( a^n \mod m \) in time \( O(\log n) \)
Primality

An integer $p$ greater than 1 is called prime if the only positive factors of $p$ are 1 and $p$.

A positive integer that is greater than 1 and is not prime is called composite.

Fundamental Theorem of Arithmetic

Every positive integer greater than 1 has a unique prime factorization

- $48 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 3$
- $591 = 3 \cdot 197$
- $45,523 = 45,523$
- $321,950 = 2 \cdot 5 \cdot 5 \cdot 47 \cdot 137$
- $1,234,567,890 = 2 \cdot 3 \cdot 3 \cdot 5 \cdot 3,607 \cdot 3,803$

Factorization

If $n$ is composite, it has a factor of size at most $\sqrt{n}$

Euclid’s theorem

There are an infinite number of primes.

Proof:

By contradiction

Suppose there are a finite number of primes: $p_1, p_2, \ldots, p_n$
Distribution of Primes

- If you pick a random number $n$ in the range $[x, 2x]$, what is the chance that $n$ is prime?

Famous Algorithmic Problems

- Primality Testing:
  - Given an integer $n$, determine if $n$ is prime
- Factoring
  - Given an integer $n$, determine the prime factorization of $n$

Factoring

- Factor the following 232 digit number [RSA768]:

  12301866845301177551304949583849627
  20772853569595334792197322452151726
  40050726365751874520219978646938995
  64749427740638459251925573263034537
  31548268507917026122142913461670429
  21431160222124047927473779408066535
  1419597459856902143413

  $\equiv$

  334780716989568987860441698482126908177047949837
  137685689124313889828837938780022876147116525317
  43087737814467999489
  $\times$

  367460436667995904282446337996279526322791581643
  430876426760322838157396665112792333734171433968
  10270092798736308917
Greatest Common Divisor

• GCD(a, b): Largest integer d such that d|a and d|b

  - GCD(100, 125) = 
  - GCD(17, 49) = 
  - GCD(11, 66) = 
  - GCD(13, 0 ) = 
  - GCD(180, 252) = 

GCD and Factoring

a = 2^3 • 3 • 5^2 • 7 • 11 = 46,200

b = 2 • 3^2 • 5^3 • 7 • 13 = 204,750

GCD(a, b) = 2^{min(3,1)} • 3^{min(1,2)} • 5^{min(2,3)} • 7^{min(1,1)} • 11^{min(1,0)} • 13^{min(0,1)}

Factoring is expensive!
Can we compute GCD(a,b) without factoring?

Useful GCD fact

If a and b are positive integers, then
\[
gcd(a, b) = gcd(b, a \mod b)
\]

Proof:
By definition \[ a = (a \div b) \cdot b + (a \mod b) \]

If d|a and d|b then d|(a \mod b):

If d|b and d|(a \mod b) then d|a :

Euclid’s Algorithm

Repeatedly use the GCD fact to reduce numbers until you get GCD(x,0)=x

GCD(660,126)
Euclid’s Algorithm

- \( \text{GCD}(x, y) = \text{GCD}(y, x \mod y) \)

```c
int GCD(int a, int b){ /* a >= b, b > 0 */
    int tmp;
    int x = a;
    int y = b;
    while (y > 0){
        tmp = x % y;
        x = y;
        y = tmp;
    }
    return x;
}
```

Example: \( \text{GCD}(660, 126) \)

Bézoit’s Theorem

If \( a \) and \( b \) are positive integers, then there exist integers \( s \) and \( t \) such that

\[
\text{gcd}(a,b) = sa + tb.
\]

Extended Euclid’s Algorithm

- Can use Euclid’s Algorithm to find \( s, t \) such that \( sa + tb = \text{gcd}(a,b) \)
- e.g. \( \text{gcd}(35,27) \):

  \[
  \begin{align*}
  35 &= 1 \cdot 27 + 8 \\
  27 &= 3 \cdot 8 + 3 \\
  8 &= 2 \cdot 3 + 2 \\
  3 &= 1 \cdot 2 + 1 \\
  2 &= 2 \cdot 1 + 0 \\
  1 &= 3 - 1 \cdot 2 = 3 - 1 (8 - 2 \cdot 3) = (-1) \cdot 8 + 3 \cdot 3 \\
  &= (-1) \cdot 8 + 3 (27 - 3 \cdot 8 ) = 3 \cdot 27 + (-10) \cdot 8 \\
  
  \end{align*}
  \]

Multiplicative Inverse mod \( m \)

Suppose \( \text{GCD}(a, m) = 1 \)

By Bézoit’s Theorem, there exist integers \( s \) and \( t \) such that \( sa + tm = 1 \).

\( s \) is the multiplicative inverse of \( a \):

\[
1 = (sa + tm) \mod m = sa \mod m
\]
Solving Modular Equations

Solving $ax \equiv b \pmod{m}$ for unknown $x$ when $\gcd(a,m)=1$.

1. Find $s$ such that $sa+tm=1$
2. Compute $a^{-1} = s \mod m$, the multiplicative inverse of $a$ modulo $m$
3. Set $x = (a^{-1} \cdot b) \mod m$

Multiplicative Cipher: $f(x) = ax \mod m$

For a multiplicative cipher to be invertible:
$f(x) = ax \mod m : \{0, m-1\} \rightarrow \{0, m-1\}$
must be one to one and onto

Lemma: If there is an integer $b$ such that $ab \mod m = 1$, then the function $f(x) = ax \mod m$ is one to one and onto.