Admin:

Pset #2 due tonight.
Pset #3 out tonight
Project proposals due 3/23

Today:

Exponentiation by repeated squaring

Multiplicative inverses mod p

Finding large primes

Euclid's gcd alg (extended)

Orders of elements

Generators

Readings:

Ferguson
Katz/Lindell
Paar/Pelzl
Smart

} all have good treatment

"Repeated squaring" to compute a^b in field

(Here b is a non-negative integer)

$$a^b = \begin{cases} 1 & \text{if } b=0 \\ (a^{b/2})^2 & \text{if } b>0,\ b \text{ even} \\ a \cdot a^{b-1} & \text{if } b \text{ odd} \end{cases}$$

Requires $\leq 2 \cdot \lg(b)$ multiplications in field (<u>efficient</u>)

$\approx$ a few milliseconds for $a^b \pmod p$ 1024-bit integers

$\approx \Theta(k^3)$ time for k-bit inputs

Computing (multiplicative) inverses:

<u>Theorem</u>: (For $GF(p)$ called "Fermat's Little Theorem")

In $GF(q)$ $(\forall a \in GF(q)^*)$ $a^{q-1} = 1$

<u>Corollary:</u> $(\forall a \in GF(q))$ $a^q = a$

<u>Corollary:</u> $(\forall a \in GF(q)^*)$ $a^{-1} = a^{q-2}$

<u>Example:</u> $3^{-1} \pmod 7$

$= 3^5 \pmod 7$

$= 5 \pmod 7$

- How to find large (k-bit) random prime #?

  <u>Generate & test:</u> <u>do</u> $p \leftarrow$ random k-bit integer
  <u>until</u> p is prime

- Works because primes are "dense":

  about $2^k/\ln(2^k)$ k-bit primes (Prime Number Theorem)

  $\Rightarrow$ one of every $\approx 0.69k$ k-bit integers is prime.

- To test if a large <u>randomly-chosen</u> k-bit integer is prime, it suffices to test

$$2^{p-1} \stackrel{?}{=} 1 \pmod{p}$$

  - This works with high probability (w.h.p) for <u>random</u> p; doesn't work for <u>adversarially</u> chosen p.
  - See CLRS for Miller-Rabin primality test (randomized)
  - Technically, above gives "base-2 pseudoprime", but this is almost always prime
  - $\exists$ deterministic poly-time primality test (Agrawal, Kayal, Saxena 2002):

    Test $(x-a)^p = x^p - a \pmod{p}$ $\quad$ x variable

    which is true <u>iff</u> p is prime

    Test mod p & mod $x^r - 1$ for small r & small a's.

    (storage requirements? see handout)

Order of elements (in Z_p^* or Z_n^*):

Define: $\text{order}_n(a)$ = "order of a, modulo n"

$= \text{least } t>0 \text{ s.t. } a^t = 1 \pmod{n}$

Recall Fermat's Little Theorem:

If p prime, then $(\forall a \in Z_p^*)\ a^{p-1} = 1 \pmod{p}$

For general n, we have Euler's Theorem:

$$(\forall n)(\forall a \in Z_n^*)\ a^{\varphi(n)} = 1 \pmod{n}$$

where $Z_n^* = \{a : \gcd(a,n) = 1\}$

= multiplicative group modulo n

$$\varphi(n) = |Z_n^*|$$

Example: $Z_{10}^* = \{1, 3, 7, 9\}$

$\varphi(10) = 4$

$3^4 = 1 \pmod{10}$

Thus $\varphi(n)$ is well-defined for all n, &

$\text{order}_n(a)$ is also well-defined.

Can we say more?

Divisors

- $d \mid a \equiv$ "d divides a" (evenly)
  $\equiv (\exists k)\ a = d \cdot k$
- d is a <u>divisor</u> of a if $d \geq 0$ & $d \mid a$
- $(\forall d)\ d \mid 0$
- $(\forall a)\ 1 \mid a$
- If d is a divisor of a & a divisor of b, then d is a <u>common divisor</u> of a & b.
- The <u>greatest common divisor</u> of a & b is the largest of their common divisors.
  [But $\gcd(0,0) = 0$ by definition.]
- <u>Examples:</u> $\gcd(24, 30) = 6$
  $\gcd(5, 0) = 5$
  $\gcd(33, 12) = 3$
- <u>Def:</u> a & b are <u>relatively prime</u> if $\gcd(a,b) = 1$

- <u>Euclid's algorithm</u> for computing $\gcd(a,b)$ $[a,b \geq 0]$:

$$\gcd(a,b) = \begin{cases} a & \text{if } b=0 \\ \gcd(b, a \bmod b) & \text{else} \end{cases}$$

- <u>Example:</u> $\gcd(7,5)$

$$\begin{aligned} &= \gcd(5,2) \\ &= \gcd(2,1) \\ &= \gcd(1,0) \\ &= 1 \end{aligned}$$

- Running time is $\approx \lg(a) \cdot \lg(b)$ bit operations

(Polynomial running time, like multiplying.)

Theorem $(\forall a,b)(\exists x,y)\ ax+by=\gcd(a,b)$

Proof "by example" $a=7,\ b=5$

$$\left.\begin{aligned} 7 &= 7\cdot 1 + 5\cdot 0 \\ 5 &= 7\cdot 0 + 5\cdot 1 \end{aligned}\right\}\ \text{initial values}$$

$$2 = 7\cdot 1 + 5\cdot(-1) \quad [\text{subtract 2 eqns}]$$

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

$$= ax + by$$

This is the "extended version of Euclid's algorithm".

Computing modular multiplicative inverses with Euclid's extended alg:

Suppose $a \in \mathbb{Z}_p^*$ (so $1 \le a < p$ & $\gcd(a,p)=1$, p prime(?))

How to compute $a^{-1} \pmod p$?

If p prime: $a^{-1} = a^{p-2} \pmod p$

Otherwise:

Find x,y s.t. $ax+py=1$

so $ax = 1 \pmod p$

and $x = a^{-1} \pmod p$

Example: $5^{-1} = 3 \pmod 7$

Example: mod $p = 7$

	1	2	3	4	5	6	7 ...	
1	1	1	1	1	1	1	1 ...	order(1) = 1
2	2	4	1	2	4	1	2 ...	order(2) = 3
3	3	2	6	4	5	1	3 ...	order(3) = 6
4	4	2	1	4	2	1	4 ...	order(4) = 3
5	5	4	6	2	3	1	5 ...	order(5) = 6
6	6	1	6	1	6	1	6 ...	order(6) = 2

Fermat

Def: $\langle a \rangle = \{a^i : i \geq 0\}$ = subgroup generated by a

Example: $\langle 2 \rangle = \{2, 4, 1\}$ (in $\mathbb{Z}_7^*$)

Theorem: $\text{order}(a) = |\langle a \rangle|$

Theorem: If p prime: $\text{order}_p(a) \mid (p-1)$.

Theorem: $|\langle a \rangle| \mid |\mathbb{Z}_n^*|$

or: $\text{order}_n(a) \mid \varphi(n)$ equivalently.

Generators

Def: If $\text{order}_p(g) = p-1$
then g is a generator of Z_p^*.
(i.e. $\langle g \rangle = Z_p^*$)

Theorem: If p is a prime and
g is a generator mod p, then
$$g^x = y \pmod p$$
has a <u>unique</u> solution x $(0 \le x < p-1)$
for each $y \in Z_p^*$.

Def: x is the "discrete logarithm"
of y, base g, modulo p.

x	=	1	2	3	4	5	6
g^x	=	3	2	6	4	5	1

for $g = 3$, modulo 7

Theorem: $\mathbb{Z}_n^*$ has a generator (i.e. $\mathbb{Z}_n^*$ is <u>cyclic</u>) iff n is

$$2,\ 4,\ p^m,\ \text{or}\ 2p^m$$

for some prime p & $m \geq 1$.

Theorem: If p is prime, the number of generators mod p is $\varphi(p-1)$

Example: $p = 11$

$\mathbb{Z}_{11}^*$ has $\varphi(10) = 4$ generators (they are 2, 6, 7, and 8).

<u>How to find a generator mod a prime p?</u>

In general, seems to require knowledge of factorization of $p-1$.

While factoring is hard, we can <u>create</u> primes for which factoring $p-1$ is trivial.

Def: If p & q are both primes &

$$p = 2q + 1$$

then p is a "safe prime" and

q is a "Sophie Germain prime".

Examples: $p = 23$, $q = 11$ $\quad$ $p = 11$, $q = 5$

$p = 59$, $q = 29$ $\quad$ ...

Theorem: If p is a safe prime

then $p - 1 = 2 \cdot q$

so $(\forall a \in \mathbb{Z}_p^*)\ \text{order}_p(a) \in \{1, 2, q, 2q\}$.

It is not hard to find safe primes. ("Probability" that a prime p is safe is $\approx 1/\ln(p)$, empirically.)

Can test if g is a generator mod $p = 2q+1$ easily:

check that $g^{p-1} = 1 \pmod p$ $\quad$ ✓ by Fermat

& $g^2 \neq 1 \pmod p$ $\quad$ $[\text{order}_p(g) \neq 2]$

& $g^q \neq 1 \pmod p$ $\quad$ $[\text{order}_p(g) \neq q]$

then $\text{order}_p(g) = p - 1$.

We can use "generate & test" again: (for "safe prime" p)

$$\underline{\text{do}} \quad g \xleftarrow{R} \mathbb{Z}_p^*$$
$$\underline{\text{until}} \quad \text{order}_p(g) = p-1$$

Generators are quite common:

<u>Theorem:</u> If $p = 2q+1$ is a "safe prime"

then # generators mod p

$$= \varphi(p-1)$$
$$= q-1 \qquad \text{(almost half of them!)}$$

(In general:

<u>Theorem:</u> If p prime, then

generators mod p

$$= \varphi(p-1)$$
$$\geq \frac{p-1}{6 \ln\ln(p-1)}$$

)

So generate & test works well for finding generators modulo a safe prime p, or modulo any prime p for which you know $\varphi(p-1)$.