MA/CSSE 473 Day 07

Extended Euclid's Algorithm

Modular Division

Fermat's little theorem intro

MA/CSSE 473 Day 07

- **Student Questions**
- Extended Euclid Algorithm, the “calculate forward, substitute backward” approach
- Modular Division
- Fermat's Little Theorem
- Intro to primality testing.

Recap: Euclid's Algorithm for gcd

```
def euclid(a, b):
    """ INPUT:  Two integers a and b with a >= b >= 0
        OUTPUT: gcd(a, b)"""
    if b == 0:
        return a
    return euclid(b, a % b)
```

Another place to read about modular arithmetic, including exponentiation and inverse: Weiss Sections 7.4-7.4.4

recap: gcd and linear combinations

- Lemma: If **d** divides both **a** and **b**, and **d** = **a**x + **b**y for some integers x and y, then **d** = gcd(**a**,**b**)
- Proof – we did it last class session.

Forward-backward Example: gcd (33, 14)

We want to find x and y such that 33x + 14y = 1

- 33 = 2*14 + 5
- 14 = 2 * 5 + 4
- 5 = 1 * 4 + 1
- 4 = 4 * 1 + 0, so gcd(33, 14) = 1.
- **Now work backwards**
- 1 = 5 - 4. Substitute 4 = 14 - 2*5:
- 1 = 5 – (14 - 2*5) = 3*5 - 14. Substitute 5 = 33 - 2*14:
- 1 = 3(33 - 2*14) -14 = 3 * 33 – 7 * 14
- Thus x = 3 and y = -7 Done!

A good place to stop and check!

Extended Euclid Algorithm

```
def euclidExtended(a, b):
    """ INPUT:  Two integers a and b with a >= b >= 0
        OUTPUT: Integers x, y, d such that d = gcd(a, b)
                and d = ax + by"""
    print ("     ", a, b) # so we can see the process.
    if b == 0:
        return 1, 0, a
    x, y, d =  euclidExtended(b, a % b)
    return y, x - a//b*y, d
```

- Proof that it works
  - I decided that it is a bit advanced for students who just saw Modular Arithmetic for the first time two sessions ago.
  - If you are interested, look up “extended Euclid proof”
  - We’ll do a convincing example.

Calculate Modular Inverse (if it exists)

- Assume that gcd(a, N) = 1. Define "inverse of a". Unique?
- The extended Euclid's algorithm gives us integers **x** and **y** such that a**x** + N**y** = 1
- This implies a**x** ≡ 1 (mod N), so **x** is the inverse of a
- **Example:** Find 14^{-1} mod 33
  - We saw before that 3*33 - 7*14 = 1
  - -7 ≡ 26 (mod 33)
  - So 14^{-1} ≡ 26 (mod 33)

Check: **14*26 = 364 = 11*33 + 1.**

- If inverse mod N exists, it is unique
- Recall that running time for Euclid's algorithm is $\Theta(k^3)$, where k is the number of bits of N.

Modular division

- We can only divide **b** by **a** (modulo **N**) if **N** and **a** are relatively prime
- In that case $b/a = b \cdot a^{-1}$
- What is the running time for modular division?

Primality Testing

- The numbers 7, 17, 19, 71, and 79 are primes, but what about 717197179 (a typical social security number)?
- There are some tricks that might help. For example:
  - If n is even and not equal to 2, it's not prime
  - n is **divisible by 3 iff** the sum of its decimal digits is divisible by 3,
  - n is **divisible by 5 iff** it ends in 5 or 0
  - n is **divisible by 7 iff** $\lfloor n/10 \rfloor$ **- 2*n%10** is divisible by 7
  - n is **divisible by 11 iff** (sum of n's odd digits) – (sum of n's even digits) is divisible by 11.
  - when checking for factors, we only need to consider prime integers as candidates
  - When checking for prime factors, we only need to examine integers up to sqrt(n)

Primality testing

- Factoring is harder than primality testing.
- Is there a way to tell whether a number is prime without actually factoring the number?

Like a few other things that we have done so far ion this course, this discussion follows Dasgupta, *et. al.*, *Algortihms* (McGraw-Hill 2008)

Fermat's Little Theorem (1640 AD)

- **Formulation 1:** If p is prime, then for every integer a with $1 \le a < p$, $a^{p-1} \equiv 1 \pmod{p}$
- **Formulation 2:** If p is prime, then for every integer a with $1 \le a < p$, $a^p \equiv a \pmod{p}$
- These are clearly equivalent.
  - How do we get from each to the other?
- We will examine a combinatorial proof of the first formulation.

Fermat's Little Theorem: Proof (part 1)

- **Formulation 1:** If **p** is prime, then for every number **a** with $1 \le \mathbf{a} < \mathbf{p}$, $\mathbf{a}^{\mathbf{p}-1} \equiv 1 \pmod{\mathbf{p}}$
- Let S = {1, 2, …, **p**-1}
- **Lemma**
  - For any nonzero integer **a**, multiplying all of the numbers in S by **a** (mod **p**) permutes S
  - I.e. $\{a \cdot n \pmod{p} : n \in S\} = S$
- **Example:** **p**=7, a=3.

i	1	2	3	4	5	6
3i	3	6	2	5	1	4

- **Proof of the lemma**
  - Suppose that $\mathbf{a} \cdot i \equiv \mathbf{a} \cdot j \pmod{\mathbf{p}}$.
  - Since **p** is prime and $\mathbf{a} \neq 0$, **a** has an inverse.
  - Multiplying both sides by $\mathbf{a}^{-1}$ yields $i \equiv j \pmod{\mathbf{p}}$.
  - Thus, multiplying the elements of S by **a** (mod **p**) takes each element to a different element of S.
  - Thus (by the pigeonhole principle), every number 1..**p**-1 is $\mathbf{a} \cdot i \pmod{\mathbf{p}}$ for some i in S.

Fermat's Little Theorem: Proof (part 2)

- **Formulation 1:** If **p** is prime, then for every number **a** with $1 \leq \mathbf{a} < p$,
  $\mathbf{a}^{\mathbf{p}-1} \equiv 1 \pmod{\mathbf{p}}$
- Let S = {1, 2, …, **p**-1}
- **Recap of the Lemma:**
  Multiplying all of the numbers in S by **a** (mod **p**) permutes S
- **Therefore:**
  $\{1, 2, \ldots, \mathbf{p}-1\} = \{\mathbf{a}\cdot 1 \pmod{\mathbf{p}}, \mathbf{a}\cdot 2 \pmod{\mathbf{p}}, \ldots \mathbf{a}\cdot(\mathbf{p}-1) \pmod{\mathbf{p}}\}$
- Take the product of all of the elements on each side .
  $(\mathbf{p}-1)! \equiv \mathbf{a}^{\mathbf{p}-1}(\mathbf{p}-1)! \pmod{\mathbf{p}}$
- Since **p** is prime, (**p**-1)! is relatively prime to **p**, so we can divide both sides by it to get the desired result:
  $\mathbf{a}^{\mathbf{p}-1} \equiv 1 \pmod{\mathbf{p}}$

Recap: Fermat's Little Theorem

- **Formulation 1:** If p is prime, then for every number a with $1 \leq a < p$, $a^{p-1} \equiv 1 \pmod{p}$
- **Formulation 2:** If p is prime, then for every number a with $1 \leq a < p$, $a^p \equiv a \pmod{p}$

Memorize this one. Know how to prove it.

Easy Primality Test?

- Is N prime?
- Pick some **a** with 1 < **a** < N
- Is $\mathbf{a}^{N-1} \equiv 1 \pmod{N}$?
- If so, N is prime; if not, N is composite
- Nice try, but…
  - Fermat's Little Theorem is <u>not</u> an "if and only if" condition.
  - It doesn't say what happens when N is <u>not</u> prime.
  - N may not be prime, but we might just happen to pick an **a** for which $\mathbf{a}^{N-1} \equiv 1 \pmod{N}$
  - **Example:** 341 is not prime (it is 11·31), but $\mathbf{2}^{340} \equiv 1 \pmod{341}$

"composite" means "not prime"

- **Definition:** We say that a number **a** **passes the Fermat test** if $\mathbf{a}^{N-1} \equiv 1 \pmod{N}$. If **a** passes the Fermat test but N is composite, then **a** is called a **Fermat liar**, and N is a **Fermat pseudoprime**.
- **We can hope that** if N is composite, then <u>many</u> values of **a** will fail the Fermat test
- It turns out that this hope is well-founded
- If any integer that is relatively prime to N fails the test, then at least half of the numbers **a** such that $1 \le a < N$ also fail it.

How many "Fermat liars"?

- If N is composite, suppose we randomly pick an **a** such that $1 \le \mathbf{a} < N$.
- If gcd(a, N) = 1, how likely is it that $\mathbf{a}^{N-1}$ is $\equiv 1 \pmod{n}$?
- If $\mathbf{a}^{N-1} \not\equiv 1 \pmod{N}$ for *any* **a** that is relatively prime to N, then this must also be true for at least half of the choices of such **a** < N.
  - Let b be some number (if any exist) that passes the Fermat test, i.e. $b^{N-1} \equiv 1 \pmod{N}$.
  - Then the number a·b fails the test:
    - $(ab)^{N-1} \equiv a^{N-1}b^{N-1} \equiv a^{N-1}$, which is not congruent to 1 mod N.
  - Diagram on whiteboard.
  - For a fixed **a**, f: b⟶ab is a one-to-one function on the set of b's that pass the Fermat test,
  - so there are at least as many numbers that fail the Fermat test as pass it.
- Continued next session …