Linear Congruences

- The equation $ax = b$ for $a, b \in \mathbb{R}$ is uniquely solvable if $a \neq 0$: $x = b/a$.
- Want to extend to the linear congruence:

$$ax \equiv b \pmod{m}, \qquad a, b \in \mathbb{Z}, m \in \mathbb{N}^+. \tag{1}$$

- If x_0 is a solution then so is $x_k := x_0 + km$, $\forall k \in \mathbb{Z}$
- ...since $km \equiv 0 \pmod{m}$.
- So, uniqueness can only be modulo m.
- How many solutions modulo 4 to $2x \equiv 2 \pmod{4}$?
- $2 \cdot 1 \equiv 2 \cdot 3 \equiv 2 \pmod{4}$.
- **Claim** If $\gcd(a, m) = 1$ then (1) has at most one solution modulo m.
- **Proof.** Suppose $r, s \in \mathbb{Z}$ are solutions of (1).
  - $\cdot \Rightarrow a(r - s) \equiv 0 \pmod{m}$
  - $\cdot \Rightarrow m \mid r - s \Rightarrow r \equiv s \pmod{m}$.

Linear Congruences cont.

- The key to finding a solution:
- $x = b/a = ba^{-1}$ where a^{-1} is the solution to $ay = 1$.
- **Claim.** Let $m \in \mathbb{N}^+$, $a \in \mathbb{Z}$. Suppose $\exists \bar{a} \in \mathbb{Z}$ s.t $a\bar{a} \equiv 1 \pmod{m}$. Then for any $b \in \mathbb{Z}$, $x = b\bar{a}$ is a solution of $ax \equiv b \pmod{m}$.
- **Proof.**

$$a(b\bar{a}) \equiv a\bar{a}b \equiv 1 \cdot b \equiv b \pmod{m}.$$

- Example: to solve $3x \equiv 4 \pmod{7}$ first find $\bar{3} \pmod{7}$:
  - · $-2 \cdot 3 \equiv -6 \equiv 1 \pmod{7} \Rightarrow -2 \equiv \bar{3} \pmod{7}$.
  - · $x = \bar{3} \cdot 4 = -2 \cdot 4 = -8$ satisfies $3x \equiv 4 \pmod{7}$.
- Does $\bar{a}$ always exist?
- Can you solve $2x \equiv 1 \pmod{4}$?
- $2 \cdot 0 \equiv 2 \cdot 2 \equiv 0 \pmod{4}$ and $2 \cdot 1 \equiv 2 \cdot 3 \equiv 2 \pmod{4}$.
- What about $2x \equiv 1 \pmod{2n}$?
- What about $\bar{2}$ modulo 3?
- When does $\bar{a}$ exist? Is it unique? How can we find it?

Inverse Modulo m

- **Theorem.** If a, m are relatively prime integers and $m > 1$ then there exists a unique inverse of a modulo m denoted as $\bar{a}$.
- **Proof.**
  - $\cdot$ $\exists s, t \in \mathbb{Z}$ s.t. $as + mt = 1$
  - $\cdot$ $\Rightarrow as \equiv 1 \pmod{m} \Rightarrow s$ is an inverse modulo m
  - $\cdot$ Since an inverse is a solution to $ax \equiv 1 \pmod{m}$ uniqueness was already proved.
- **Cor.** $\bar{a}$ is given by the extended Euclid algorithm.
- $\cdot$ Example: $\gcd(3, 7) = 1 \Rightarrow \exists\bar{3}$ modulo 7
  $\cdot$ $7 = 2\cdot 3+1 \Rightarrow -2\cdot 3+7 = 1 \Rightarrow \bar{3} \equiv -2 \pmod{7}$.

The Chinese Remainder Theorem

- Example. Pick an integer $n \in [0, 104]$.
  - Tell me its remainders modulo 3, 5, and 7 (r_3, r_5, r_7).
  - Let me "guess": $n = 70r_3 + 21r_5 + 15r_7 \bmod 105$.
- **Def.** $m_1, \ldots, m_n$ are pairwise relatively prime if $\forall i, j$, $\gcd(m_i, m_j) = 1$.
- **Theorem.** Let $m_1, \ldots, m_n \in \mathbb{N}^+$ be pairwise relatively prime. The set of equations

$$x \equiv a_i \pmod{m_i} \qquad i = 1, 2 \ldots n \tag{2}$$

  has a unique solution modulo $M := \prod_1^n m_i$.
- Comments:
  - If x is a solution then so is $x + kM$ for any $k \in \mathbb{Z}$.
  - There exists a unique solution $x \in \mathbb{N} \cap [0, M - 1]$.
- Example: a solution to
  - $x \equiv r_3 \pmod 3, x \equiv r_5 \pmod 5, x \equiv r_7 \pmod 7$
  - is $x = 70r_3 + 21r_5 + 15r_7 \bmod 105$.
  - **The key:**
  - $70 \bmod 3 = 1$, $70 \bmod 5 = 0$, $70 \bmod 7 = 0$
  - $21 \bmod 3 = 0$, $21 \bmod 5 = 1$, $21 \bmod 7 = 0$
  - $15 \bmod 3 = 0$, $15 \bmod 5 = 0$, $15 \bmod 7 = 1$

Proof of the CRT

- **Proof.** A solution exists if $\exists x_i$, $i = 1, \ldots, n$ s.t.:

$$\begin{aligned} x_i &\equiv 1 \pmod{m_i} \\ x_i &\equiv 0 \pmod{m_j} \quad \forall j \neq i. \end{aligned} \tag{3}$$

· Indeed, $x := \sum_1^n a_j x_j$ satisfies

$$x \equiv \sum_{j=1}^{n} a_j (x_j \bmod m_i) \equiv \sum_{j=1}^{n} a_j \delta_{ij} \equiv a_i \pmod{m_i}.$$

· We prove (3) constructively: let $s_i = M/m_i$.

· Then, $s_i = \prod_{i \neq j} m_j \equiv 0 \pmod{m_j}$ for $j \neq i$.

· s_i has an inverse modulo m_i, $\bar{s_i}$

· ...since $\gcd(m_i, s_i) = 1$. Let $x_i := s_i \bar{s_i}$.

· For $i = 1, \ldots, n$, x_i satisfies (3).

· In our example:
$$\begin{aligned} s_3 &= 5 \cdot 7 = 35 \ , \ \bar{s_3} = 2 \\ s_5 &= 3 \cdot 7 = 21 \ , \ \bar{s_5} = 1 \\ s_7 &= 3 \cdot 5 = 15 \ , \ \bar{s_7} = 1. \end{aligned}$$

· Uniqueness: suppose x and y satisfy (2).

· $\Rightarrow x - y \equiv 0 \pmod{m_i}$ for $i = 1, \ldots, n$.

· The next lemma completes the proof:

· **Lemma.** If $m_i \in \mathbb{N}^+$ are pairwise relatively prime and $m_i \mid s$ for $i = 1, \ldots, n$ then $\prod_1^n m_i \mid s$.

Proof of the CRT cont.

- **Lemma.** If $m_i \in \mathbb{N}^+$ are pairwise relatively prime and $m_i \mid s$ for $i = 1, \ldots, n$ then $\prod_1^n m_i \mid s$.
- **Proof.** By induction on n.
  - · For $n = 1$ the statement is trivial.
  - · Assuming it holds for $n = N$ we want to prove it for $n = N + 1$.
  - · Let $a := \prod_1^N m_i$ and let $b = m_{N+1}$.
  - · $\exists l \in \mathbb{Z}$ s.t. $s = la$
  - · ... by the inductive hypothesis $a \mid s$.
  - · $b \mid s \Rightarrow b \mid l$
  - · ... because a and b are relatively prime.
  - · $\Rightarrow \exists k \in \mathbb{Z}$ s.t. $l = bk$.
  - · $\Rightarrow s = al = abk \Rightarrow ab \mid s$.

Computer Arithmetic with Large Integers

- Want to work with *very* large integers.
- Choose $m_1, \ldots, m_n$ pairwise relatively prime.
- To compute $N_1 + N_2$ or $N_1 \cdot N_2$:

$$N_i \longleftrightarrow (N_i \bmod m_1, \ldots, N_i \bmod m_n)$$
$$N_1 + N_2 \longleftrightarrow (N_1 + N_2 \bmod m_1, \ldots, N_1 + N_2 \bmod m_n)$$
$$N_1 \cdot N_2 \longleftrightarrow (N_1 \cdot N_2 \bmod m_1, \ldots, N_1 \cdot N_2 \bmod m_n)$$

- The lhs of the last two equation can readily be computed component wise.
- Requires efficient transition:

$$N \longleftrightarrow (N \bmod m_1, \ldots, N \bmod m_n).$$

- Advantages:
  - · Allows arithmetic with very large integers.
  - · Can be readily parallelized.
- Example. The following are pairwise relatively prime.

  $\{m_i\}_1^5 = \{2^{35} - 1, 2^{34} - 1, 2^{33} - 1, 2^{29} - 1, 2^{23} - 1\}$
- We can add and multiply positive integers up to $M = \prod_1^5 m_i > 2^{184}$.

Fermat's Little Theorem

- If p is a prime and $p \nmid a \in \mathbb{Z}$ then $a^{p-1} \equiv 1 \pmod{p}$. Moreover, for any $a \in \mathbb{Z}$, $a^p \equiv a \pmod{p}$.
- **Proof.** Let $A = \{1, 2, \ldots, p-1\}$, and let
  - $B = \{1a \bmod p, 2a \bmod p, \ldots, (p-1)a \bmod p\}$.
  - $0 \notin B$ so $B \subset A$.
  - $|A| = p - 1$ so if $|B| = p - 1$ then $A = B$.
  - Let $1 \leq i \neq j \leq p - 1$, then
  - $ia \bmod p \neq ja \bmod p$
  - $\iff ia \not\equiv ja \pmod{p}$
  - $\iff p \nmid a(i - j)$
  - $\Rightarrow A = B$.

$$\Rightarrow (p-1)! = \prod_{i=1}^{p-1} (ia \bmod p) \equiv a^{p-1}(p-1)! \pmod{p}.$$

  - $\Rightarrow a^{p-1} \equiv 1 \pmod{p}$
  - …since $\gcd((p-1)!, p) = 1$.
  - In particular, $a^p \equiv a \pmod{p}$.
  - The latter clearly holds for a s.t. $p \mid a$ as well.

Private Key Cryptography

- Alice (aka A) wants to send an encrypted message to Bob (aka B).
- A and B might share a private key known only to them.
- The same key serves for encryption and decryption.
- Example: Caesar's cipher $f(m) = m + 3 \bmod 26$.
  - `ABCDEFGHIJKLMNOPQRSTUVWXYZ`
  - `WKH EXWOHU GLG LW`
  - `THE BUTLER DID IT`
  - Note that $f(m) - 3 \bmod 26 = m$
- Slightly more sophisticated: $f(m) = am + b \bmod 26$
  - Example: $f(m) = 4m + 1 \bmod 26$
  - ...oops $f(0) = f(13) = 1$.
  - Decryption: solve for m, $(am + b) \bmod 26 = c$, or $am \equiv c - b \pmod{26}$.
  - Need $\exists \bar{a}$, or $\gcd(a, 26) = 1$.
  - Weakness of this cipher: suppose the triplet `QMB` is much more popular than all other triplets...

Private Key cont.

- However, some private key systems are totally immune to non-physical attacks:
  - A and B share the only two copies of a long list of random integers s_i for $i = 1, \ldots, N$.
  - A sends B the message $\{m_i\}_{i=1}^{n}$ encrypted as:
  - $c_i = m_i + s_{K+i} \bmod 26$ for $i = 1, \ldots, n$.
  - A also sends the key K and deletes $s_{K+1}, \ldots, s_{K+n}$.
  - B decrypts A's message by computing
  - $c_i - s_{K+i} \bmod 26$.
  - Upon decryption B also deletes $s_{K+1}, \ldots, s_{K+n}$.
  - Pros: bullet proof cryptography system
  - Cons: horrible logistics
- Cons (any private key system):
  - Only predetermined users can exchange messages

Public Key Encryption

- A uses B's public encryption key to send an encrypted message to B.
- Only B has the decryption key that allows decoding of messages encrypted with his public key.
- BIG advantage: A need not know nor trust B.

RSA

- **Generating the keys.**
  - · Choose two very large (hundreds of digits) primes p, q.
  - · Let $n = pq$.
  - · Choose $e \in \mathbb{N}$ relatively prime to $(p-1)(q-1)$.
  - · Compute d, the inverse of e modulo $(p-1)(q-1)$.
- Publish the modulos n and the encryption key e.
- Keep the decryption key d to yourself.
- **Encryption protocol.**
  - · The message is divided into blocks each represented as $M \in \mathbb{N} \cap [0, n-1]$. Each block M is encrypted:

$$C = M^e \pmod{n}.$$

- Example. Encrypt "stop" using $e = 13$ and $n = 2537$:
  - · `s t o p` $\longleftrightarrow$ `18 19 14 15` $\longleftrightarrow$ `1819 1415`
  - · $1819^{13} \bmod 2537 = 2081$ and $1415^{13} \bmod 2537 = 2182$ so
  - · `2081 2182` is the encrypted message.
  - · We did not need to know $p = 43, q = 59$ for that.
  - · By the way, $\gcd(13, 42 \cdot 58) = 1$.

RSA cont.

- **Decryption:** compute $C^d \bmod n$.
- **Claim.** $C^d \bmod n = M$.
- **Lemma** Suppose p is prime. Then for $a \in \mathbb{Z}$
  - $\cdot\ p \nmid a$ and $k \equiv 0 \pmod{p-1} \Rightarrow a^k \equiv 1 \pmod{p}$.
  - $\cdot\ m \equiv 1 \pmod{p-1} \Rightarrow a^m \equiv a \pmod{p}$.
- **Proof of Claim.**
  - $\cdot\ ed \equiv 1 \pmod{p-1}$ and $ed \equiv 1 \pmod{q-1}$
  - $\cdot$ …since $ed \equiv 1 \pmod{(p-1)(q-1)}$
  - $\cdot \Rightarrow M^{ed} \equiv M \pmod{p}$, and $M^{ed} \equiv M \pmod{q}$.
  - $\cdot \Rightarrow M^{ed} \equiv M \pmod{n}$.
  - $\cdot \Rightarrow M^{ed} \bmod n = M$.
  - $\cdot \Rightarrow C^d \bmod n = [M^e \bmod n]^d \bmod n$

$$= M^{ed} \bmod n$$
$$= M.$$

- **Proof of lemma.**
  - $\cdot\ k = l(p-1)$ for some $l \in \mathbb{Z}$.
  - $\cdot \Rightarrow a^k = \left(a^{p-1}\right)^l \equiv \left(a^{p-1} \bmod p\right)^l \equiv 1 \pmod{p}$.
  - $\cdot$ If $p \mid a$, $a^m \equiv a \pmod{p}$ for *any* m.
  - $\cdot$ If $p \nmid a$, use $m - 1 \equiv 0 \pmod{p-1}$ above.

Probabilistic Primality Testing

- RSA requires really large primes.
- The popular way of testing primality is through probabilistic algorithms.
- The procedure for randomized testing of n's primality is based on a readily computable test $\mathrm{T}(b,n)$: is $b \in \mathbb{Z}_n^* := \{1,\ldots,n-1\}$ a "witness" for n's primality.
- Example. Is $b^{n-1} \equiv 1 \pmod{n}$?
- The answer is always positive if n is prime.
- Unfortunately, the answer might be positive even if n is composite: $2^{340} \equiv 1 \pmod{341}$ and $341 = 11 \cdot 31$.
- The probability that a randomly chosen b will be a witness to the "primality" of the composite n, depends on T.
- Machine Learning: false positive rate of T on n, $\mathrm{FP}(n)$.
- Need to control the overall FP rate of T: establish a lower bound q, on the probability of a false witness for *any* n.
- If m randomly chosen bs have all testified that n is prime then the probability that n is composite $\leq q^m$.

Probabilistic Primality Testing cont

IsPrime($\mathbf{n}, \varepsilon, [\mathbf{T}, q]$): Primality Testing
Input: $n \in \mathbb{N}^+$ - the prime suspect
$\varepsilon \in (0, 1)$ - probability of false classification
T - a particular prime test
FPr - a lower bound on Prob(false witness)
Output: "yes, with probability $\geq 1 - \varepsilon$", or "no"

$Pr = 1$
while $Pr > \varepsilon$
  randomly draw $b \in \mathbb{Z}_n^* := \{1, 2 \ldots, n - 1\}$
  if T(b, n)
    $Pr := Pr \cdot$ FPr
  else
    return "no"
return "yes, with probability $\geq 1 - \varepsilon$"

- For a given ε, the complexity clearly depends on FPr, the false positive rate of T.
- How many false witnesses b can there possibly be?

Fermat's Pseudoprimes

- **Def.** If n is a composite and $b^{n-1} \equiv 1 \pmod{n}$ then n is a *Fermat pseudoprime* to the base b.
- Let T_F be the Fermat test and assume n is composite.
- n is a Fermat pseudoprime to the base b if and only if $\mathrm{T}_F(b, n)$ is a FP.
- What is the probability, q_n, that $\mathrm{T}_F(b, n)$ yields a FP for a randomly chosen $b \in \mathbb{Z}_n^* := \{1, 2, \ldots, n-1\}$?
- If $k = |\{b \in \mathbb{Z}_n^* : \mathrm{T}_F(b, n) \text{ is positive}\}|$, for k out of the $n-1$ possible bs, $\mathrm{T}_F(b, n)$ gives a FP.
- Since each of the bs is equally likely to be drawn, $q_n = k/(n-1)$.
- Are there composites n which are Fermat pseudoprimes to *relatively* many bases b?

Carmichael numbers

- **Def.** A composite n which is a Fermat pseudoprime for any b with $\gcd(n, b) = 1$ is a *Carmichael number*.
- Example. $n = 561$ is a Carmichael number.
  - · Suppose $b \in \mathbb{Z}_n^*$ with $\gcd(b, n) = 1$.
  - · $n = p_1 p_2 p_3$ with $p_1 = 3$, $p_2 = 11$, $p_3 = 17$.
  - · Check: $n - 1 \equiv 0 \pmod{p_i - 1}$ for $i = 1, 2, 3$.
  - · $\Rightarrow b^{n-1} \equiv 1 \pmod{p_i}$ for $i = 1, 2, 3$
  - · ...since $p_i \nmid b$.
  - · $\Rightarrow b^{n-1} \equiv 1 \pmod{n}$.
- T_F can perform miserably on Carmichael numbers: it will yield a FP for most bs.
- Example. If $n = p_1 p_2 p_3$ is a Carmichael numbers then

$$\begin{aligned} 1 - q_n &\leq \frac{n/p_1 - 1}{n - 1} + \frac{n/p_2 - 1}{n - 1} + \frac{n/p_3 - 1}{n - 1} \\ &\leq \frac{1}{p_1} + \frac{1}{p_2} + \frac{1}{p_3} \end{aligned}$$

- Aside: Use of a Carmichael number instead of a prime factor in the modulus of an RSA cryptosystem is likely to make the system fatally vulnerable - Pinch (97).

The Rabin-Miller Test

- **Input:**
  - $\cdot$ $n = 2^s t + 1$ where t is odd and $s \in \mathbb{N}$
  - $\cdot$ $b \in \mathbb{Z}_n^*$
- $\mathbf{T_{RM}}$**:** Does *exactly* one of the following hold?
  - $\cdot$ $b^t \equiv 1 \pmod{n}$ or
  - $\cdot$ $b^{2^j t} \equiv -1 \pmod{n}$ for one $0 \leq j \leq s-1$.
- **Claim.** If n is prime, $\mathrm{T}_{RM}(b, n)$ is positive $\forall b \in \mathbb{Z}_n^*$.
- **Fact.** If n is composite the FP rate is at most 1/4.
- The probability that a composite n will survive m tests $\mathrm{T}_{RM}(b, n)$ with randomly chosen bs is $\leq 4^{-m}$.
- The claim is a corollary of the following lemma.
- **Lemma.** If $p \neq 2$ is prime and $p \mid b^{2^s t} - 1$ then p divides exactly one factor in

$$b^{2^s t} - 1 = (b^t - 1)(b^t + 1)(b^{2t} + 1) \ldots (b^{2^{s-1} t} + 1).$$

- Note that in our case $p = 2^s t + 1$ so for b relatively prime to p, $p \mid b^{2^s t} - 1$ by Fermat's theorem.
- **Sketch of lemma's proof.**

· Induction on s, base is trivial.

· $p \mid b^{2^s t} - 1 \Rightarrow p \mid (b^{2^{s-1}t} - 1)(b^{2^{s-1}t} + 1)$.

· But p cannot divide both factors since then

· $p \mid (b^{2^{s-1}t} + 1) - (b^{2^{s-1}t} - 1) = 2$.

Pseudorandom Numbers

- For the randomized algorithms we need a random number generator.
- Most languages provide you with a function "rand".
- There is nothing random about such a function...
- Being deterministic it creates pseudorandom numbers.
- Example. The linear congruential method.
  - · Choose a modulus $m \in \mathbb{N}^+$,
  - · a multiplier $a \in \{2, 3, \ldots, m-1\}$ and
  - · an increment $c \in \mathbb{Z}_m := \{0, 1, \ldots, m-1\}$.
  - · Choose a seed $x_0 \in \mathbb{Z}_m$ (time is typically used).
  - · Compute $x_{n+1} = ax_n + c \pmod{m}$.
- Warning: a poorly implemented rand(), such as in C, can wreak havoc on Monte Carlo simulations.

Database 101

- Problem: How can we efficiently store, retrieve and delete records from a large database?
- For example, students records.
- Each record has a unique key (e.g. student ID).
- Shall we keep an array sorted by the key?
- Easy retrieval but difficult insertion and deletion.
- How about a table with an entry for every possible key?
- Often infeasible, almost always wasteful.

Hashing

- Store the records in an array of size N.
- N should be somewhat bigger than the expected number of records.
- The location of a record is given by $h(k)$ where k is the key and h is the *hashing function* which maps the space of keys to $\mathbb{Z}_N$.
- Example: $h(k) := k \bmod N$.
- A collision occurs when $h(k_1) = h(k_2)$ and $k_1 \neq k_2$.
- To minimize collisions makes sure N is sufficiently large.
- You can re-hash the data if the table gets too full.
- A good hashing function should distribute the images of the possible set of keys fairly evenly over $\mathbb{Z}_N$.
- Ideally, $P(h(k) = i) = 1/N$ for any $i \in \mathbb{Z}_N$.
- When collisions occur there are mechanisms to resolve them (buckets, next empty cell, etc.)

Tentative Prelim Coverage

IMPORTANT: The only type of calculator that you can bring with you to the prelim is one *without any memory or programming capability*. If you have any doubt about whether or not your calculator qualifies it probably doesn't but feel free to ask one of the professors.

- Chapter 0:
  - Sets
    - Set builder notation
    - Operations: union, intersection, complementation, set difference
  - Relations:
    - reflexive, symmetric, transitive, equivalence relations
    - transitive closure
  - Functions
    - Injective, surjective, bijective
    - Inverse function
  - Important functions and how to manipulate them:
    - exponent, logarithms, ceiling, floor, mod, polynomials

- Summation and product notation
- Matrices (especially how to multiply them)
- Proof and logic concepts
  - logical notions ($\Rightarrow$, $\equiv$, $\neg$)
  - Proofs by contradiction

- Chapter 1
  - You do not have to write algorithms in their notation
  - You must be able to *read* algorithms in their notation

- Chapter 2
  - induction vs. strong induction
  - guessing the right inductive hypothesis
  - inductive (recursive) definitions

- Number Theory - everything we covered in class including
  - Fundamental Theorem of Arithmetic
  - gcd, lcm
  - Euclid's Algorithm and its extended version
  - Modular arithmetics, linear congruences, modular inverse

  - CRT
  - Fermat's little theorem
  - RSA
  - Probabilistic primality testing
- Chapter 4:
  - Section 4.1, 4.2, 4.3
  - Sum and product rule