Wrap of Number Theory & Midterm Review

- Primes, GCD, and LCM (Section 3.5 in text)
- Midterm Review
  - Sections 1.1-1.7
    - Propositional logic
    - Predicate logic
    - Rules of inference and proofs
  - Sections 2.1-2.3
    - Sets and Set operations
    - Functions
  - Sections 3.4-3.5
    - Integers, div, mod, congruence, applications
    - Primes and their properties

Recall: Fundamental Theorem of Arithmetic

Fundamental Theorem of Arithmetic

- FTA Theorem. $\forall n \in Z^+$ where n > 1, n is a prime or a product of primes in nondecreasing order. (Proof in a later section)
- In other words, primes are the "building blocks" of integers
- FTA examples:
  - $50 = 2 \times 5 \times 5 = 2^1 \cdot 5^2$
  - $72 = 2 \times 2 \times 2 \times 3 \times 3 = 2^3 3^2$
  - $5 = 5^1$

Testing whether a number is prime

- Naïve algorithm for primality testing:
  - Input n:

    For a = 2,…, n-1:Test whether a | n.

    If no a divides n, then n prime.
- Is there a better (faster) algorithm?
  - Do we need to test all the numbers from 2 to n-1?

Testing whether a number is prime

- Thm: *n* composite $\rightarrow$ *n* has a prime factor $\leq \sqrt{n}$
  - Proof: *n* composite $\rightarrow \exists$a (1<a<n) n = ab for some integer b > 1.
    Suppose a > $\sqrt{n}$ and b > $\sqrt{n}$.
    Then ab > $\sqrt{n} \cdot \sqrt{n}$ i.e., ab > n.
    This contradicts ab = n. Therefore, a $\leq \sqrt{n}$ or b $\leq \sqrt{n}$.
    If a or b is prime, we are done. Otherwise, by FTA, a is product of prime factors < a and b is product of prime factors < b. Therefore, *n* has a prime factor $\leq \sqrt{n}$. QED.
- Corollary: If n does not have a prime factor $\leq \sqrt{n}$, then n is prime

Algorithm for Primality

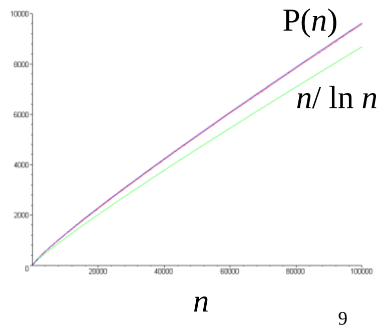
Algorithms for Primality and Prime Factorization

- Algorithm for Primality: Given n, test whether any *prime* from 2 to $\sqrt{n}$ divides n. If none does, then n is prime.
  - Example: Is 311 a prime? Test 2, 3, 5, 7, 11, 13, 17 $\leq \sqrt{311}$
    None divides 311, therefore 311 is a prime. (Note: only tested 7 numbers instead of the 309 numbers in the naïve algorithm!)
- Algorithm for prime factorization of n: Find prime factors $p_1 \leq \sqrt{n},\ p_2 \leq \sqrt{n/p_1},\ p_3 \leq \sqrt{n/(p_1 p_2)} \ldots$
- Example: Find prime factorization of 819
  819→ Test 2, 3,..→3 | 819, so $p_1 = 3$; Next, 819/3 = 273
  273→Test 2, 3,…→ 3 | 273, so $p_2 = 3$; Next, 273/3 = 91
  91 →Test 2, 3, 5, 7…→ 7 | 91, so $p_3 = 7$; Next, 91/7 = 13 (a prime)
  Therefore, 819 = 3·3 ·7·13

Ain’t primal enuff for me, mate!

How many primes are there?

- Euclid's theorem (circa 300 BC): There are infinitely many primes.
  - Proof by contradiction: See text.
  - Corollary: For any positive integer *n*, there is always a prime greater than *n*.
- How many primes ≤ *n*?
  - Let P(*n*) = number of primes ≤ *n*.
  - Prime Number Theorem: P(*n*) is approximately *n*/ ln *n* as *n* grows without bound.
  - Cor.: Probability that a random positive int. ≤ *n* is prime = (*n*/ ln *n*)/*n* = 1/ ln *n*



Greatest Common Divisor (GCD)

- Example:
  - Positive divisors of 16 = 1, 2, 4, 8, 16
  - Positive divisors of 24 = 1, 2, 3, 4, 6, 8, 12
  - Greatest Common Divisor gcd(16,24) = 8
- For any nonzero a,b ∈ Z, gcd(a,b) = largest integer d such that d | a and d | b
  - gcd(10,15) = 5, gcd(7,15) = 1
  - a, b are relatively prime iff gcd(a,b) = 1. E.g., 7 and 15.
- Computing gcd(a,b): Use prime factorization of a, b

$$a = p_1^{a_1} p_2^{a_2} \ldots p_n^{a_n}\ ,\ b = p_1^{b_1} p_2^{b_2} \ldots p_n^{b_n}\ (a_i, b_i \text{ can be } 0)$$

$$\gcd(a,b) = p_1^{\min(a_1,b_1)} p_2^{\min(a_2,b_2)} \ldots p_n^{\min(a_n,b_n)}$$

$$\text{E.g. } 60 = 2^2 3 \cdot 5,\ 72 = 2^3 3^2, \gcd(60,72) = 2^2 3 \cdot 5^0 = 12$$

Least Common Multiple (LCM)

- Example:
  - Multiples of 6 = 6, 12, 18, 24, 30, …
  - Multiples of 8 = 8, 16, 24, 32, …
  - Least Common Multiple lcm(6,8) = 24
- For any a,b $\in$ Z^+, lcm(a,b) = smallest c $\in$ Z^+ such that a | c and b | c.
  - lcm(4,6) = 12, lcm(5,10) = 10, lcm(5,11) = 55
- Computing lcm(a,b): Use prime factorization of a, b

$$a = p_1^{a_1} p_2^{a_2} \ldots p_n^{a_n}, \; b = p_1^{b_1} p_2^{b_2} \ldots p_n^{b_n} \; (a_i, b_i \text{ can be } 0)$$

$$\text{lcm}(a,b) = p_1^{\max(a_1,b_1)} p_2^{\max(a_2,b_2)} \ldots p_n^{\max(a_n,b_n)}$$

$$\text{E.g. } 6 = 2 \cdot 3, \; 8 = 2^3, \text{lcm}(6,8) = 2^3 \cdot 3 = 24$$

- Theorem: gcd(a,b)·lcm(a,b)=ab

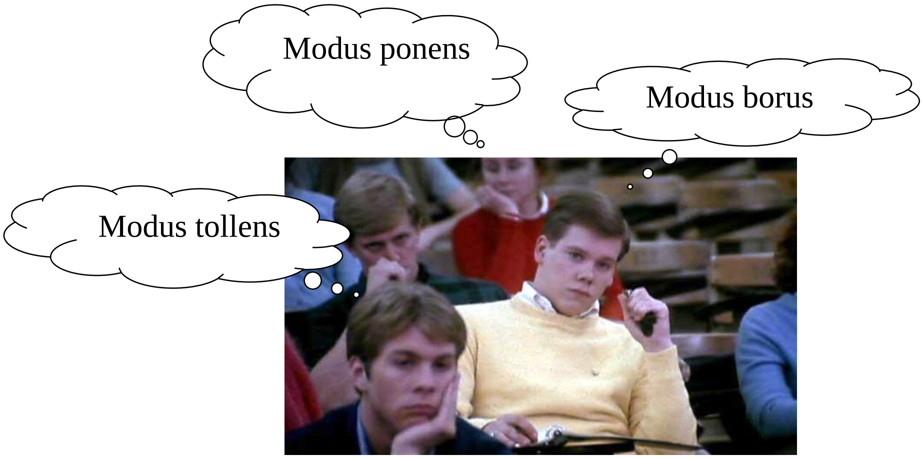
Midterm Review: Chapter 1 (Sections 1.1-1.7)

- Propositional Logic
  - Propositions, logical operators $\neg$, $\wedge$, $\vee$, $\oplus$, $\rightarrow$, $\leftrightarrow$, truth tables for operators, precedence of logical operators
  - Compound propositions, truth tables for compound propositions
  - Converse, contrapositive, and inverse of p $\rightarrow$ q
  - Converting from/to English and propositional logic
- Propositional Equivalences
  - Tautology versus contradiction
  - Logical equivalence p $\equiv$ q
  - Tables of logical equivalences (tables 6, 7, 8 in text)
  - De Morgan’s laws
  - Showing two compound propositions are logically equivalent via (a) truth table method and (b) via equivalences in tables 6, 7, 8.

Predicate Logic

- Predicates and Quantifiers
  - Predicates, variables, and domain of each variable
  - Universal and existential quantifiers ∀ and ∃ (uniqueness ∃!)
  - Truth value of a quantifier statement
  - Logical equivalence of two quantified statements
  - Negation and De Morgan's laws for quantifiers
  - Translating to/from English
- Nested Quantifiers
  - Translating to/from English, negating nested quantifiers

Rules of Inference



Rules of Inference

- Rule of inference = valid argument form. Table 1 (p. 66).
  - Modus ponens: $[p \wedge (p \rightarrow q)] \rightarrow q$
  - Modus tollens: $[(p \rightarrow q) \wedge \neg q] \rightarrow \neg p$
  - Hypothetical Syllogism: $[(p \rightarrow q) \wedge (q \rightarrow r)] \rightarrow (p \rightarrow r)$
  - Disjunctive Syllogism: : $[(p \vee q) \wedge \neg p] \rightarrow q$
  - Addition, Simplification, Conjunction
  - Resolution: $[(p \vee q) \wedge (\neg p \vee r)] \rightarrow (q \vee r)$
- Using rules of inference to prove statements from premises
- Rules of inference for quantified statements: instantiation and generalization

Proofs and Proof Methods

- Direct proof of $p \rightarrow q$: Assume p is true; show q is true.
- Proof of $p \rightarrow q$ by contraposition: Assume $\neg q$ and show $\neg p$.
- Vacuous and Trivial Proofs of $p \rightarrow q$
- Proof by contradiction of a statement p: Assume p is not true and show this leads to a contradiction ($r \wedge \neg r$).
- Proofs of equivalence for $p \leftrightarrow q$: Show $p \rightarrow q$ and $q \rightarrow p$
- Proof by cases and Existence proofs

Chapter 2: Sets and Operations (Sections 2.1-2.2)

- Sets
  - Set builder notation, set equality, Venn diagrams
  - Sets Z, Z^+, R, Q, N, ∅, singleton sets
  - Subset and proper subset
  - Cardinality, finite and infinite sets, Power set
  - Tuples, Cartesian product, truth set of a predicate
- Set operations
  - ∪, ∩, difference, complement
  - Set identities (similar to logical equivalences)
  - Proving two sets are equal: Two methods
    - Show each set is a subset of the other, OR
    - Use logical equivalences
- Bit string representation of sets and bitwise operations

Chapter 2: Functions (Section 2.3)

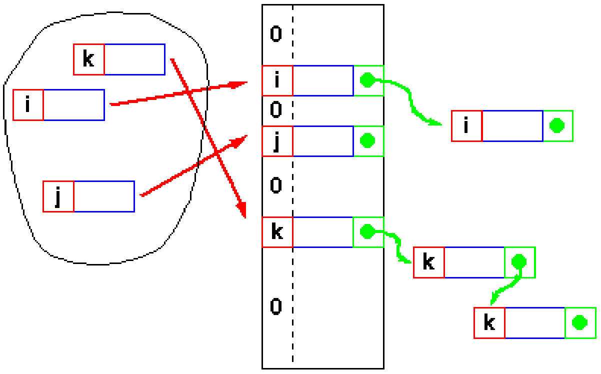
- Definition of a function
  - Domain, co-domain, range, image, preimage
  - 1-1 and onto functions, bijections
    - Know definitions and how to show 1-1, onto, or bijection
  - Inverse of a function and composition of functions
  - floor and ceiling functions
    - Know definitions and how to compute

Chapter 3: Integers and Division (Section 3.4)

- Division
  - Know definitions of a | b, factor, multiple
  - Prove identities involve |
  - Division algorithm
    - Know the statement, **div**, **mod**
- Modular arithmetic
  - Know definition and theorems
    a ≡ b (mod m) iff m | (a-b) iff a **mod** m = b **mod** m iff a = b + km

Applications of Modular Arithmetic

- Hashing
  - Hashing function
  - Collision



Applications of Modular Arithmetic

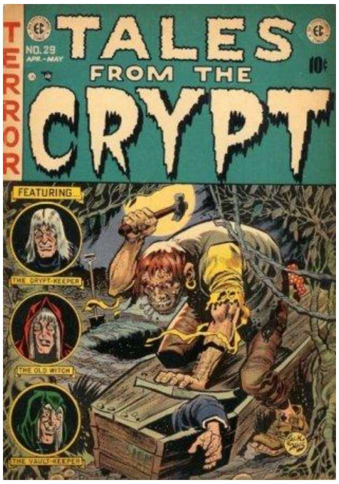
Pseudorandom numbers using linear congruential generator

$$X_{n+1} = (aX_n + b) \bmod m$$

Applications of Modular Arithmetic

Cryptology

- Caeser’s cipher
- Shift cipher
- Encryption
- Decryption



Chapter 3: Primes and GCD (Section 3.5)

- Primes
  - Definition, Fundamental Theorem of Arithmetic (FTA)
  - Algorithms for testing primality and prime factorization
  - Euclid's infinitude of primes theorem
  - Prime number theorem: Number of primes not exceeding n is approximately n / ln n as n grows without bound
- GCD and LCM
  - Definition of gcd and lcm, definition of relatively prime
  - Finding gcd and lcm through prime factorizations (using min/max of exponents)

Good luck on the midterm

- You can bring one 8 1/2" x 11" review sheet (double-sided ok, handwritten or typed but no magnifying aids please!).
- Calculators okay to use but won't really need it.

Don't sweat it!

- Go through the homeworks, lecture notes, and examples in the text
- Do the practice midterm on the website and avoid being surprised!