MA/CSSE 473
Day 9

Primality Testing

Encryption Intro

MA/CSSE 473 Day 09

- Quiz
- Announcements
- Exam coverage
- Student questions
- Review: Randomized Primality Testing.
- Miller-Rabin test
- Generation of large prime numbers
- Introduction to RSA cryptography

Exam 1 resources

- No books, notes, electronic devices (except a calculator that is not part of a phone, etc.), no earbuds or headphones.
- I will give you the Master Theorem and the formulas from Appendix A of Levitin.
- A link to an old Exam 1 is on Day 14 of the schedule page.

Exam 1 coverage

- HW 1-5
- Lectures through today
- Readings through Chapter 3.
- There is a lot of "sink in" time before the exam.
- But of course we will keep looking at new material.

Exam 1

- If you want additional practice problems for Tuesday's exam:
  - The "not to turn in" problems from various assignments
  - Feel free to post your solutions in a Piazza discussion forum and ask your classmates if they think it is correct
- Allowed for exam:
  Calculator
- See the exam specification document, linked from the exam day on the schedule page.

About the exam

- Mostly it will test your understanding of things in the textbook and things we have discussed in class or that you have done in homework.
- Will not require a lot of creativity (it's hard to do much of that in 50 minutes).
- Many short questions, a few calculations.
  - Perhaps some T/F/IDK questions (example: 5/0/3)
- You may bring a calculator.
- I will give you the Master Theorem and the formulas from Levitin Appendix A.
- Time may be a factor!
- First do the questions you can do quickly

Possible Topics for Exam - 2016

- Formal definitions of O, Θ, Ω.
- Recurrences, Master Theorem
- Fibonacci algorithms and their analysis
- Efficient numeric multiplication
- Proofs by induction (ordinary, strong)
- Extended Binary Trees
- Trominoes
- Other HW problems (assigned and suggested)
- Mathematical Induction
- Modular multiplication, exponentiation
- Extended Euclid algorithm
- Modular inverse
- What would Donald (Knuth) say?
- Binary Search
- Binary Tree Traversals
- Basic Data Structures (Section 1.4)
- Graph representations

Possible Topics for Exam - 2016

- Brute Force algorithms
- Selection sort
- Insertion Sort
- Amortized efficiency analysis
- Analysis of growable array algorithms
- Binary Search
- Binary Tree Traversals
- Basic Data Structures (Section 1.4)
- Graph representations
- BFS, DFS,
- DAGs & topological sort

Recap: Where are we now?

- For a moment, we pretend that Carmichael numbers do not exist.
- If N is prime, $a^{N-1} \equiv 1 \pmod{N}$ for all $0 < a < N$
- If N is not prime, then $a^{N-1} \equiv 1 \pmod{N}$ for at most half of the values of $a<N$.
- $\Pr(a^{N-1} \equiv 1 \pmod{N}$ if N is prime$) = 1$
  $\Pr(a^{N-1} \equiv 1 \pmod{N}$ if N is composite$) \leq ½$
- How to reduce the likelihood of error?

The algorithm (modified)

- To test N for primality
  - Pick positive integers $a_1, a_2, \ldots, a_k < N$ at random
  - For each a_i, check for $a_i^{N-1} \equiv 1 \pmod{N}$
    - Use the Miller-Rabin approach, (next slides) so that Carmichael numbers are unlikely to thwart us.
    - If a_i^{N-1} is not congruent to 1 (mod N), or Miller-Rabin test produces a non-trivial square root of 1 (mod N)
      - return false
  - return true

Does this work?

Note that this algorithm may produce a "false prime", but the probability is very low if k is large enough.

Miller-Rabin test

- A **Carmichael number** N is a composite number that passes the Fermat test for all **a** with $1 \leq$ **a** $<N$ and gcd(a, N)=1.
- **A way around the problem (Rabin and Miller):**
  (Not just for Carmichael numbers).
  Note that for some t and u (u is odd), $N-1 = 2^t u$.
- As before, compute a^{N-1}(mod N), but do it this way:
  - Calculate a^u (mod N), then repeatedly square, to get the sequence
    a^u (mod N), a^{2u} (mod N), …, $a^{2^t u}$ (mod N) $\equiv a^{N-1}$ (mod N)
- Suppose that at some point, $a^{2^i u} \equiv 1$ (mod N), but $a^{2^{i-1} u}$ is not congruent to 1 or to N-1 (mod N)
  - then we have found a nontrivial square root of 1 (mod N).
  - We will show that if 1 has a nontrivial square root (mod N), then N cannot be prime.

Example (first Carmichael number)

- N = 561. We might randomly select a = 101.
  - Then $560 = 2^4 \cdot 35$, so u=35, t=4
  - $a^u \equiv 101^{35} \equiv 560$ (mod 561) which is -1 (mod 561)
    (we can stop here)
  - $a^{2u} \equiv 101^{70} \equiv 1$ (mod 561)
  - …
  - $a^{16u} \equiv 101^{560} \equiv 1$ (mod 561)
  - So 101 is not a witness that 561 is composite (we can say that 101 is a ***Miller-Rabin liar*** *for 561,* if indeed 561 is composite)
- Try a = 83
  - $a^u \equiv 83^{35} \equiv 230$ (mod 561)
  - $a^{2u} \equiv 83^{70} \equiv 166$ (mod 561)
  - $a^{4u} \equiv 83^{140} \equiv 67$ (mod 561)
  - $a^{8u} \equiv 83^{280} \equiv 1$ (mod 561)
  - So 83 is a witness that 561 is composite, because 67 is a non-trivial square root of 1 (mod 561).

Lemma: Modular Square Roots of 1

- If there is an s which is neither 1 or -1 (mod N), but $s^2 \equiv 1$ (mod N), then N is not prime
- **Proof** (by contrapositive)**:**
  - Suppose that N is prime and $s^2 \equiv 1$ (mod N)
  - $s^2-1 \equiv 0$ (mod N) [subtract 1 from both sides]
  - $(s - 1)(s + 1) \equiv 0$ (mod N) [factor]
  - So N divides (s - 1) (s + 1) [def of congruence]
  - Since N is prime, N divides (s - 1) or N divides (s + 1) [def of prime]
  - s is congruent to either 1 or -1 (mod N) [def of congruence]
- This proves the lemma, which validates the Miller-Rabin test

Accuracy of the Miller-Rabin Test

- Rabin* showed that if N is composite, this test will demonstrate its non-primality for at least ¾ of the numbers **a** that are in the range 1…N-1, even if **N** is a Carmichael number.
- Note that 3/4 is the worst case; randomly-chosen composite numbers have a much higher percentage of witnesses to their non-primeness.
- If we test several values of **a**, we have a very low chance of incorrectly flagging a composite number as prime.

*Journal of Number Theory 12 (1980) no. 1, pp 128-138

Efficiency of the Test

- Testing a k-bit number is $\Theta(k^3)$
- If we use the fastest-known integer multiplication techniques (based on Fast Fourier Transforms), this can be pushed to $\Theta(k^2 * \log k * \log \log k)$

Testing "small" numbers

- **From Wikipedia article on the Miller-Rabin primality test:**
- When the number N we want to test is small, smaller fixed sets of potential witnesses are known to suffice. For example, Jaeschke* has verified that
  - if N < 9,080,191, it is sufficient to test a = 31 and 73
  - if N < 4,759,123,141, it is sufficient to test a = 2, 7, and 61
  - if N < 2,152,302,898,747, it is sufficient to test a = 2, 3, 5, 7, 11
  - if N < 3,474,749,660,383, it is sufficient to test a = 2, 3, 5, 7, 11, 13
  - if N < 341,550,071,728,321, it is sufficient to test a = 2, 3, 5, 7, 11, 13, 17

* Gerhard Jaeschke, "On strong pseudoprimes to several bases", Mathematics of Computation 61 (1993)

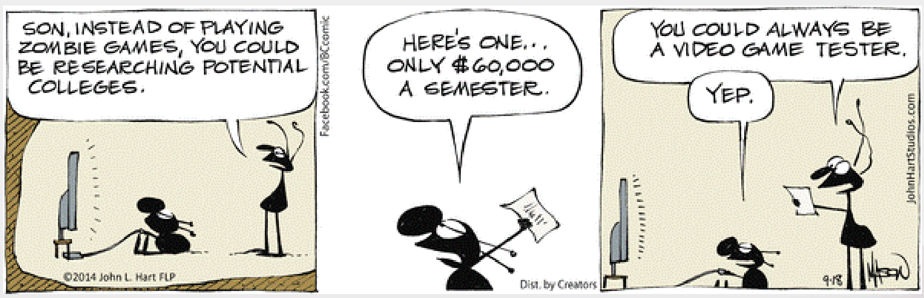
Generating Random Primes

- For cryptography, we want to be able to quickly generate random prime numbers with a large number of bits
- Are prime numbers abundant among all integers? Fortunately, yes
- Lagrange's prime number theorem
  - Let $\pi(N)$ be the number of primes that are $\leq$ N, then $\pi(N) \approx N / \ln N$.
  - Thus the probability that an k-bit number is prime is approximately $(2^k / \ln (2^k)) / 2^k \approx 1.44 / k$

Random Prime Algorithm

- To generate a random k-bit prime:
  - Pick a random k-bit number N
  - Run a primality test on N
  - If it passes, output N
  - Else repeat the process
  - Expected number of iterations is $\Theta(k)$

Interlude


SON, INSTEAD OF PLAYING ZOMBIE GAMES, YOU COULD BE RESEARCHING POTENTIAL COLLEGES.
Facebook.com/BCcomic
©2014 John L. Hart FLP
HERE'S ONE... ONLY $60,000 A SEMESTER.
Dist. by Creators
YOU COULD ALWAYS BE A VIDEO GAME TESTER.
YEP.
JohnHartStudios.com
9-18