CS F364
Design & Analysis of Algorithms

PROBLEM DOMAIN – NUMBER THEORY

Testing for Primes:

- Quadratic Residues
- Rabin-Miller algorithm
    - Error Bounds and Time Complexity

3/5/2015 Sundar B. CSIS, BITS, Pilani

Quadratic Residues

Quadratic Residue Theorem :

- *For odd prime* ***p*** *and* ***e>=1****, the equation $x^2 = 1 \pmod{\mathbf{p}^{\mathbf{e}}}$ has* <u>*only two solutions*</u>*:*
  - $\mathbf{x = 1 \pmod{p^e}}$ and $\mathbf{x = -1 \pmod{p^e}}$

Proof:

- $\mathbf{x^2 = 1 \pmod{p^e}}$ implies $\mathbf{p^e \mid (x-1)(x+1)}$
- Since **p>2**,
  - we may have $\mathbf{p \mid (x-1)}$ or $\mathbf{p \mid (x+1)}$ but not both
- If $\mathbf{p \nmid x-1}$ then $\mathbf{gcd(p^e, x-1) = 1}$ and so $\mathbf{p^e \mid (x+1)}$
- Similarly, if $\mathbf{p \nmid x+1}$ then $\mathbf{gcd(p^e, x+1) = 1}$ and so $\mathbf{p^e \mid (x-1)}$
- Thus $\mathbf{x = 1 \pmod{p^e}}$ or $\mathbf{x = -1 \pmod{p^e}}$

Quadratic Residues

- A number x is <u>*a square root of 1 modulo n*</u>:
  - if it satisfies the equation $x^2 = 1 \pmod{n}$.
    - 1 (mod n) and -1 (mod n) are referred to as <u>*trivial square roots*</u>
- A number x is <u>a *non-trivial square root of 1 modulo n*</u>:
  - if it satisfies the equation $x^2 = 1 \pmod{n}$ and
  - it is neither 1 (mod n) nor (-1 mod n).

Quadratic Residues

- Corollary (to Quadratic Residue Theorem):
  - If there exists a nontrivial square root of 1, *modulo n*, then n is composite.
- Proof:
  - n != 1 (obvious)
  - n != 2 (easy to verify)
  - By contra-positive of the (quadratic residue) theorem:
    - *if there exists a nontrivial square root of 1, modulo n, then n is not an odd prime.*

Primality Testing – Approach II

Idea:

- *Instead of trying just one potential witness for compositeness*
  - i.e. a in Z^*_n not satisfying Fermat congruence

  we try multiple potential witnesses and we use both Fermat congruence and non-trivial square roots.

Primality Testing – Approach II: Steps and Correctness

Algorithm (outline):

1. Let a be a random witness

   i.e. chosen randomly in $Z_n \setminus \{ 0 \}$
2. Let k and d be such that

   $n-1 = 2^k * d$ where d is odd and $k>=1$,
3. Compute $b_i = a \wedge (2^i * d) \bmod n$, for each $i = 0$ to k
   1. If $b_k \neq 1$ then n is composite
   2. ...

This is correct by Fermat's Theorem because:

$b_k = a^{n-1} \pmod{n}$

PRIMALITY TESTING – APPROACH II - STEPS AND CORRECTNESS [2]

Algorithm (outline):

1. Let a be a random witness

   i.e. chosen randomly in $Z_n \setminus \{ 0 \}$

2. Let k and d be such that

   $n-1 = 2^k * d$ where d is odd and $k>=1$,

3. Compute $b_i = a \wedge (2^i * d) \bmod n$, for each $i = 0$ to k
   1. If $b_k \ != 1$ then n is composite
   2. If for any i, $b_i = 1$ but $b_{i-1} \ != 1$ and $b_{i-1} \ != n-1$ then n is composite

This is correct by the Corollary to the Quadratic Residue Theorem because:

$$b_i = (b_{i-1})^2$$

PRIMALITY TESTING – APPROACH II – A MONTE-CARLO ALGORITHM

procedure WITNESS(a,n)

// n is odd and a is chosen randomly from Z_n and

1. d=n-1; k=0;
2. while (d mod 2 == 0) { /* Invariant: $\underline{n-1 = 2^k * d}$ */
   d = d / 2; k = k + 1;
   }
3. $b_0 = a^d$ mod n
4. for i = 1 to k {
   1. $b_i = b_{i-1} * b_{i-1}$ (mod n);
   2. if (b_i == 1 && b_{i-1} != 1 && b_{i-1} != n-1) return 1;

   }
4. if b_k != 1 return 1;
5. return 0; // n is prime

k steps

Multiplication of two m-bit numbers takes **m*m** time (naive algorithm)

k steps

- Running time $= k * m^2 = \Theta((\log n)^3)$
  - because $k <= \lfloor \log_2 n \rfloor$ and $m = \lfloor \log_2 n \rfloor$

Primality Testing – Approach II – A Monte-carlo algorithm

- Claim (w.o. proof): *The probability that WITNESS fails to produce a witness for an odd composite n is at most 1/4.*
- Claim : *WITNESS successfully produces a witness for any odd composite n even if n is a Carmichael number.*
  - Proof: *<u>Corrollary of the Quadratic Residue Theorem</u> applies <u>for all composite numbers</u> – including Carmichael numbers.*

Primality Testing – Miller-rabin

- **Miller-Rabin Algorithm**

```
MR(n,s)  {  // s is the number of trials
 for k = 1 to s {
     a = random(1,n-1);
     if  (WITNESS(a,n)) return "composite";
  }
 return "prime";
 }
```

Running Time: $\Theta(s(\log n)^3)$

Claim:

- **Miller-Rabin** is a polynomial time Monte-Carlo algorithm with 1-way error :
  - *if it returns composite then it is correct* and
  - *if it returns prime then it errs with probability at most* $\boldsymbol{4^{-s}}$

MILLER-RABIN (MR): NUMBER OF TRIALS

Estimating the required number of trials of Miller-Rabin:

- Consider a number N randomly chosen (with a fixed bit length)
  - Let **A** denote the event that N is prime and
  - Let **B** denote the event **MR**(N,s) returns “prime”.
    - We must estimate **Pr[A/B]**

ASIDE: CONDITIONAL PROBABILTIY

- Bayes's Theorem (for events A and B):
  - Pr[A/B] = (Pr[A]*Pr[B/A])/Pr[B]
- Since
  - B = (B ∩ A) ∪ (B ∩ A') and
  - B ∩ A and B ∩ A' are mutually exclusive
  - Pr[B] = Pr[B∩A] + Pr[B∩A']
  - Pr[B] = Pr[A]*Pr[B/A] + Pr[A']*Pr[B/A']
- Now substituting Pr[B] into *Bayes's Theorem*:
  - Pr[A/B] = (Pr[A]*Pr[B/A])/(Pr[A]*Pr[B/A]+Pr[A']*Pr[B/A'])

Aside: Conditional Probability

- We can now estimate
  - the probability that
    - <u>*a number N is prime*</u> (i.e. event **A**)
  - given
    - <u>*MR(N,s) returns "prime"*</u> (i.e. event **B**)
- using the result from previous slide
  - Pr[A/B] = (Pr[A]*Pr[B/A])/(Pr[A]*Pr[B/A]+Pr[A']*Pr[B/A'])

ASIDE: CONDITIONAL PROBABILITY

- For computing
  - Pr[A/B] = (Pr[A]*Pr[B/A])/(Pr[A]*Pr[B/A]+Pr[A’]*Pr[B/A’])
- we must compute:
  - Pr[A], the probability that N is prime
  - Pr[B/A], the probability MR(N,s) returns “prime” when N is prime
  - Pr[B/A’], the probability MR(N,s) returns “prime” when N is composite

Estimating the Probability MR(n,s) Yields The Correct Result

- Pr[A], *the probability that N is prime*

  $= 1 / \ln(N)$ by Prime Number Theorem

- Pr[B/A], *the probability MR(N,s) returns "prime"* when N is prime

  $= 1$

- Pr[B/A'], *the probability MR(N,s) returns "prime"* when N is composite

  $= 4^{-s}$

Miller-Rabin (MR): Number of Trials

Estimating the required number of trials of MR:

- Pr[A/B]
- >= (Pr[A]*Pr[B/A]) / (Pr[A]*Pr[B/A]+Pr[A']*Pr[B/A'])
- >= (1/ln(N)) / ((1/ln(N)) + (1-(1/ln(N)))*4^{-s})
- >= 1 / (1 + (ln(N)-1)*4^{-s})
- >= 1/2 for $s >= \log_4(\ln(N)-1)$

- i.e. if we run **MR(N,s)** for $s>= \log_4(\ln(N)-1)$ and it returns "prime",
  - then **N** is prime with probability > 1/2
- Let us call this **s** the **half-life** of **N** and denote it as $\mathbf{s_{half}}$ (**N**) :
  - *if **MR(N,s)** returns "prime"* for some **s** > $\mathbf{s_{half}(\log_2 N)}$
  - then <u>**N** is prime with probability</u> **>= 1 /(1 + 4^(s_{half}(N) – s))**

MILLER-RABIN (MR): NUMBER OF TRIALS - EXAMPLE

- For example, for a 1024 bit number N,
  - $s_{half}(N) = \log_4(\ln(N)-1) = \log_4(\log_2(N)/\log_2(e) - 1)$
  - $\approx \log_4(1024/1.443) \approx 5$
    - i.e. *if MR(N, 5) returns "prime"*

    then N is prime with probability >= 1/2
- For s=50
  - $Pr[A/B] >= 1 / (1 + 4^{-45}) \approx 1$ for most practical purposes

Miller-Rabin (MR): Trials : Growth Rate

- Consider a 2m-bit number and an m-bit number
  - i.e. N^2 and N
  - $\mathbf{s}_{\mathbf{half}}(N^2) = \log_4(\ln(N^2)-1) = \log_4(\log_2(N^2)/\log_2(e) - 1)$
  - $\approx \log_4(\log_2(N^2)/\log_2(e))$
  - $= \log_4(2*\log_2(N)/\log_2(e))$
  - $= \log_4 2 + \mathbf{s}_{\mathbf{half}}(N)$
  - $= 0.5 + \mathbf{s}_{\mathbf{half}}(N)$
- We can generalize this by induction :
  - $\mathbf{s}_{\mathbf{half}}(N^{2^\wedge k}) = k/2 + \mathbf{s}_{\mathbf{half}}(N)$
  - i.e. the half-life grows slowly with respect to N
    - in fact sub-linearly with respect to logN
      - i.e. w.r.t. the size of N