Primality Testing

Sachin Tripathi

IIT(ISM), Dhanbad

Introduction

- Suppose we have an integer of 200 digits that we want to test for Primality.
- There are around 4×10^{97} primes less that 10^{100}.
- This is significantly more that the number of particles in the universe.
- Now, if the computer can handle 10^{9} primes per second, the calculation would take around 10^{81} years.

Generating Primes

- Several mathematicians attempt to develop a formula that could generates primes.
- If p is a prime number then, $M_p = 2^p - 1$
- A number in the form $M_p = 2^p - 1$ is called a Mersenne number.
- All numbers are not a prime.

Example

- $M_2 = 2^2 - 1 = 3$
- $M_3 = 2^3 - 1 = 7$
- $M_5 = 2^5 - 1 = 31$
- $M_7 = 2^7 - 1 = 127$
- $\mathbf{M_{11} = 2^{11} - 1 = 2047\ (23 \times 89)}$
- $M_{13} = 2^{13} - 1 = 8191$
- $M_{17} = 2^{17} - 1 = 131071$

Fermat's Prime

$$F_n = 2^{2^n} + 1$$

- If n is an integer then,
- $F_2 = 17$
- $F_3 = 257$
- $F_4 = 65537$
- $F_5 = 4294967297$ is not a prime (641×6700417)

Euler's Prime

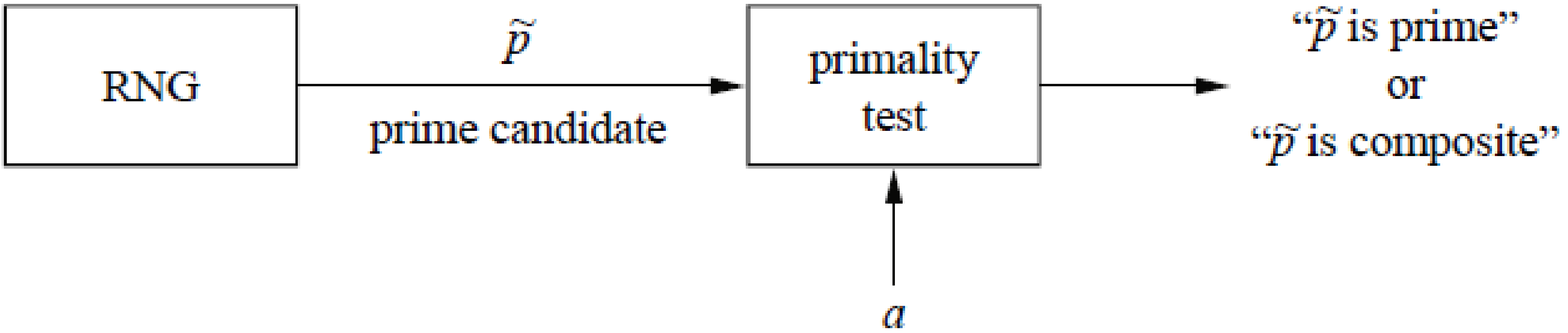
- If n is an integer then Euler's prime generating polynomial P(n)is defined as follows:

  $P(n) = n^2+n+41$

- $P(1) = 43$
- $P(2) = 47$

Finding Large Primes in RSA

- The general approach is to generate large integers at random which are then checked for primality.



Principal approach to generating primes for RSA

Divisibility Algorithm

Problem Statement: Given a number n, how can we determine if n is a prime?

Answer: The answer is that we need to see if the number is divisible

Algorithm 9.1 *Pseudocode for the divisibility test*

```
Divisibility_Test (n)                    // n is the number to test for primality
{
  r ← 2
  while (r < √n )
  {
   if (r | n) return "a composite"
   r ← r + 1
  }
   return "a prime"
}
```

Probabilistic Algorithms

- Most of the efficient algorithms for Primality testing are category of probabilistic algorithm.
- A probabilistic algorithm does not guarantee the correctness of the result.
- However, we can make the probability of error so small that it is almost certain that the algorithm has returned a correct answer.

Primality Tests

- Practical primality tests behave somewhat unusually: if the integer *p in question* is being fed into a primality test algorithm, then the answer is either

1. *"p is composite" (i.e., not a prime), which is always a true statement, or*
2. *"p is prime", which is only true with a high probability.*

Common Practice

- If the algorithm output is "composite", the situation is clear: The integer in question is not a prime and can be discarded.
- If the output statement is "prime", *p is probably* a prime. In rare cases, it yields an incorrect positive answer.
- Practical primality tests are *probabilistic algorithms*.

Fermat Primality Test

- Let $n>1$ be an integer. Choose a random integer a with $1<a<n-1$. If $a^{n-1} \neq 1 \bmod n$ then n is composite. Otherwise n is probably prime.

Fermat Primality Test
Input: prime candidate $\tilde{p}$ and security parameter s
Output: statement "$\tilde{p}$ is composite" or "$\tilde{p}$ is likely prime"
Algorithm:

```
1      FOR i = 1 TO s
1.1        choose random a ∈ {2,3,...,p̃−2}
1.2        IF a^(p̃−1) ≢ 1
1.3            RETURN ("p̃ is composite")
2      RETURN ("p̃ is likely prime")
```

- There are certain composite integers which behave like primes in the Fermat test for many values of a.
- For example 561 ($3 \times 11 \times 17$) passes the Fermat test.
- These are the **Carmichael numbers**. A Carmichael number must be the product of at least three distinct primes.
- Such special composites are very rare. For instance, there exist approximately only 100,000 Carmichael numbers below 10^{15}.

Basic Principle. *Let n be an integer and suppose there exist integers x and y with $x^2 \equiv y^2 \pmod{n}$, but $x \not\equiv \pm y \pmod{n}$. Then n is composite. Moreover, $\gcd(x-y, n)$ gives a nontrivial factor of n.*

Proof. Let $d = \gcd(x-y, n)$. If $d = n$ then $x \equiv y \pmod{n}$, which is assumed not to happen. Suppose $d = 1$. A basic result on divisibility is that if $a|bc$ and $\gcd(a, b) = 1$, then $a|c$ (see Exercise 7 in Chapter 3). In our case, since n divides $x^2 - y^2 = (x-y)(x+y)$ and $d = 1$, we must have that n divides $x+y$, which contradicts the assumption that $x \not\equiv -y \pmod{n}$. Therefore, $d \neq 1, n$, so d is a nontrivial factor of n. $\square$

Example. Since $12^2 \equiv 2^2 \pmod{35}$, but $12 \not\equiv \pm 2 \pmod{35}$, we know that 35 is composite. Moreover, $\gcd(12-2, 35) = 5$ is a nontrivial factor of 35.

Square Root Primality Test

- In Modular Arithmetic ,if n is a prime , the square root of 1 is **either +1 or -1.**
- If n is composite ,the square root is **+1 or -1 but there may be other roots.**

If n is a prime, $\sqrt{1} \bmod n = \pm 1$.
If n is a composite, $\sqrt{1} \bmod n = \pm 1$ and possibly other values.

- What are the square roots of 1 mod n if n is 7 (a prime)?
- Solution: The only square roots are 1 and −1.

$$
\begin{array}{ll}
1^2 = 1 \bmod 7 & (-1)^2 = 1 \bmod 7 \\
2^2 = 4 \bmod 7 & (-2)^2 = 4 \bmod 7 \\
3^2 = 2 \bmod 7 & (-3)^2 = 2 \bmod 7
\end{array}
$$

- What are the square roots of 1 mod n if n is 8 (a composite)?
- Solution: There are four solutions: 1, 3, 5, and 7 (which is −1).

$$
\begin{array}{ll}
1^2 = 1 \bmod 8 & (-1)^2 = 1 \bmod 8 \\
3^2 = 1 \bmod 8 & 5^2 = 1 \bmod 8
\end{array}
$$

- What are the square roots of 1 mod n if n is 22 (a composite)?
- Solution: Surprisingly, there are only two solutions, +1 and −1, although 22 is a composite.

$$1^2 = 1 \bmod 22$$
$$(-1)^2 = 1 \bmod 22$$

Contd...

- Although this test can tell us if a number is composite, it is difficult to do the testing.
- Given a number n, all numbers less than n (except 1 and n-1) must be squared to be sure that none of them is 1.
- This test can be used for a number (not +1 or -1) that when squared in modulus n has the value 1.
- This fact helps in the Miller-Rabin test.

Miller-Rabin PrimalityTest

- Let $n > 1$ be an odd integer. Write $n\text{-}1 \equiv 2^k m$ with m odd.
- Choose a random integer α with $1 < \alpha < n\text{-}1$.
- Compute $b_0 \equiv \alpha^m \pmod{n}$.If $b_0 \equiv \pm 1 \pmod{n}$,then stop and declare that n is probably prime.
- Otherwise, let $b_1 \equiv b_0^2 \pmod{n}$. If $b_1 \equiv 1 \pmod{n}$ then n is composite (and gcd $(b_0\text{-}1, n)$ gives a nontrivial factor of n).
- If $b_1 \equiv -1 \pmod{n}$ then stop and declare that n is probably prime.
- Otherwise, let $b_2 \equiv b_1^{\ 2} \pmod{n}$. If $b_2 \equiv 1 \pmod{n}$ then n is composite .
- If $b_2 \equiv -1 \pmod{n}$ then stop and declare that n is probably prime.
- Continue in this way until stopping or reaching $b_{k\text{-}1}$.if $b_{k\text{-}1} \not\equiv -1$ (mod n) ,then n is composite.

Algorithm 9.2 *Pseudocode for Miller-Rabin test*

```
Miller_Rabin_Test (n, a)                    // n is the number; a is the base.
{
    Find m and k such that n − 1 = m × 2^k
    T ← a^m mod n
    if (T = ± 1)   return "a prime"
    for (i ← 1 to k − 1)                    // k − 1 is the maximum number of steps.
    {
        T ← T^2 mod n
        if (T = +1) return "a composite"
        if (T = −1) return "a prime"
    }
    return "a composite"
}
```

Example

- Let n = 561.Then n-1 = 560 = 16*35,so $2^k = 2^4$ and m = 35 .Let $\alpha = 2$ then

$$b_0 \equiv 2^{35} \equiv 263 \pmod{561}$$

$$b_1 \equiv b_0^2 \equiv 166 \pmod{561}$$

$$b_2 \equiv b_1^2 \equiv 67 \pmod{561}$$

$$b_3 \equiv b_2^2 \equiv 1 \pmod{561}$$

since $b_3 \equiv 1 \pmod{561}$. we conclude that 561 is composite and $\gcd(b_2-1,561) = 33$,which is a non trivial factor of 561.

Thank You