Cryptography, winter 2014/15

Primality testing

Dr. Daniel Loebenberger

Theorem

There are infinitely many primes.

Fermat's Theorem

Let N be prime and $x \in \mathbb{Z}_N$ with $x \neq 0$. Then $x^{N-1} = 1$ in $\mathbb{Z}_N$.

ALGORITHM. Fermat test.

Input: A number $N \in \mathbb{Z}$ with $N \geq 2$.
Output: Either "N is composite", or "N is possibly prime".

1. $x \xleftarrow{\text{🎲}} \{1, \ldots, N-1\}$.
2. $g \leftarrow \gcd(x, N)$. If $g \neq 1$, then Return "N is composite".
3. $y \leftarrow x^{N-1}$ in $\mathbb{Z}_N$.
4. If $y \neq 1$, then Return "N is composite".
5. Return "N is possibly prime".

ALGORITHM. Strong pseudoprimality test.

Input: An odd number $N \in \mathbb{Z}$ with $N \geq 2$.
Output: Either "N is composite", or "N is probably prime".

1. $x \xleftarrow{\text{🎲}} \{1, \ldots, N-1\}$.
2. If $\gcd(x, N) \neq 1$ then Return "N is composite".
3. Write $N - 1 = 2^e m$, where m is odd.
4. $y \leftarrow x^m$ in $\mathbb{Z}_N$.
5. If $y = 1$ then Return "N is probably prime".
6. For i from 0 to $e - 1$ do steps 7.-8.
7. If $y = -1$ then Return "N is probably prime",
8. otherwise $y \leftarrow y^2$ in $\mathbb{Z}_N$.
9. Return "N is composite".

N	$N-1=2^e \cdot m$	y_0	y_1	y_2	y_3	y_4
553	$2^3 \cdot 69$	407	302	512	22	
557	$2^2 \cdot 139$	556	1			
$561 = 3 \cdot 11 \cdot 17$	$2^4 \cdot 35$	56	331	116	67	1

Table : Testing $553, 557$, and 561.

Lemma

N is a Carmichael number if and only if N is squarefree and $p-1$ divides $N-1$ for all prime divisors p of N.

Theorem

The *strong pseudoprimality test* has the following properties.

(i) If N is prime, the test returns "probably prime".

(ii) If N is composite, the test returns "composite" with probability at least $1/2$.

(iii) For an n-bit input N, the test uses $O(n^3)$ bit operations.

Theorem

The strong pseudoprimality test, repeated t times independently, has the following properties on input N.

(i) If it outputs "N is composite", then N is composite.

(ii) If it outputs "N is probably prime", then N is prime with probability at least $1 - 2^{-t}$.

ALGORITHM. Finding a pseudoprime.

Input: An integer n and a confidence parameter t.
Output: A pseudoprime number N in the range from $2^{(n-1)/2}$ to $2^{n/2}$.

1. $x \longleftarrow 2^{(n-1)/2}$.
2. Repeat steps 3 and 4 Until some N is accepted.
3. $N \xleftarrow{\text{🎲}} \{\lceil x \rceil, \ldots, \lfloor \sqrt{2}x \rfloor\}$.
4. Call the strong pseudoprimality test with input N for t independently chosen $x \xleftarrow{\text{🎲}} \{1, \ldots, N-1\}$. Return N if and only if all these tests return "probably prime". Goto step 3 if at least one of the tests answers "composite".
5. Return N.

Prime Number Theorem

For the prime counting function $\pi(x)$, the function $\vartheta(x) = \sum_{p \le x} \ln p$ and the n-th prime p_n, we have approximately

$$\pi(x) \approx \frac{x}{\ln x}, \quad \vartheta(x) \approx x, \quad p_n \approx n \ln n,$$

and more precisely

$$\frac{x}{\ln x}\left(1 + \frac{1}{2\ln x}\right) < \pi(x) < \frac{x}{\ln x}\left(1 + \frac{3}{2\ln x}\right) \text{ for } x \ge 59,$$

$$\frac{3x}{5\ln x} < \pi(2x) - \pi(x) < \frac{7x}{5\ln x} \text{ for } x \ge 21,$$

$$n\left(\ln n + \ln\ln n - \frac{3}{2}\right) < p_n < n\left(\ln n + \ln\ln n - \frac{1}{2}\right) \text{ for } n \ge 20,$$

$$x\left(1 - \frac{1}{2\ln x}\right) < \vartheta(x) < x\left(1 + \frac{1}{2\ln x}\right) \text{ for } x \ge 563.$$

Theorem

On input $n \geq 11$ and t, the output of the Algorithm *Finding a pseudoprime* is prime with probability at least $1 - 2^{-t+1}\, n$. It uses an expected number of $O(tn^4)$ bit operations.