Primality Tests Based on Fermat's Little Theorem

Manindra Agrawal

IIT Kanpur

ICDCN, IIT Guwahati

Overview

1. Fermat's Little Theorem
2. Primality Testing
3. Solovay-Strassen Algorithm
4. Miller-Rabin Algorithm
5. AKS Algorithm

Outline

1. Fermat's Little Theorem
2. Primality Testing
3. Solovay-Strassen Algorithm
4. Miller-Rabin Algorithm
5. AKS Algorithm

Fermat's Little Theorem

Pierre de Fermat (1601-1665)

Theorem

If n is prime then for every a, $1 \leq a < n$, $a^{n-1} = 1 \ (mod\ n)$.

Proof

- Consider the sequence of numbers $a * 1 \ (mod\ n)$, $a * 2 \ (mod\ n)$, ..., $a * (n-1) \ (mod\ n)$ for any $1 \leq a < n$.
- None of these are zero, and no pair is equal:
  - Follows from the primality of n.
- Therefore,

$$\prod_{i=1}^{n-1} a * i = \prod_{i=1}^{n-1} i \ (mod\ n).$$

- Canceling $\prod_{i=1}^{n-1} i$ from both sides we get

$$a^{n-1} = 1 \ (mod\ n).$$

Proof

- Consider the sequence of numbers $a * 1 \ (mod\ n)$, $a * 2 \ (mod\ n)$, $\ldots$, $a * (n-1) \ (mod\ n)$ for any $1 \leq a < n$.
- None of these are zero, and no pair is equal:
  - Follows from the primality of n.
- Therefore,

$$\prod_{i=1}^{n-1} a * i = \prod_{i=1}^{n-1} i \ (mod\ n).$$

- Canceling $\prod_{i=1}^{n-1} i$ from both sides we get

$$a^{n-1} = 1 \ (mod\ n).$$

Proof

- Consider the sequence of numbers $a * 1 \ (mod\ n)$, $a * 2 \ (mod\ n)$, ..., $a * (n-1) \ (mod\ n)$ for any $1 \leq a < n$.
- None of these are zero, and no pair is equal:
  - Follows from the primality of n.
- Therefore,

$$\prod_{i=1}^{n-1} a * i = \prod_{i=1}^{n-1} i \ (mod\ n).$$

- Canceling $\prod_{i=1}^{n-1} i$ from both sides we get

$$a^{n-1} = 1 \ (mod\ n).$$

Proof

- Consider the sequence of numbers $a * 1 \pmod n$, $a * 2 \pmod n$, ..., $a * (n-1) \pmod n$ for any $1 \le a < n$.
- None of these are zero, and no pair is equal:
  - Follows from the primality of n.
- Therefore,

$$\prod_{i=1}^{n-1} a * i = \prod_{i=1}^{n-1} i \pmod n.$$

- Canceling $\prod_{i=1}^{n-1} i$ from both sides we get

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

Consequences

- Fermat's Little Theorem identifies a crucial property of prime numbers.
- Instrumental in design of some of the most important primality testing algorithms.

Consequences

- Fermat's Little Theorem identifies a crucial property of prime numbers.
- Instrumental in design of some of the most important primality testing algorithms.

Outline

1. Fermat's Little Theorem
2. Primality Testing
3. Solovay-Strassen Algorithm
4. Miller-Rabin Algorithm
5. AKS Algorithm

The Problem

Given a number n, decide if it is prime efficiently.

By efficiently, one means an algorithm taking $\log^{O(1)} n$ steps.

The Problem

Given a number n, decide if it is prime efficiently.

By efficiently, one means an algorithm taking $\log^{O(1)} n$ steps.

School Method

Try dividing by all numbers $< n$ or better, $\leq \sqrt{n}$.

Takes time $\Omega(\sqrt{n}) = \Omega(2^{\frac{1}{2}\log n})$.

School Method

Try dividing by all numbers $< n$ or better, $\leq \sqrt{n}$.

Takes time $\Omega(\sqrt{n}) = \Omega(2^{\frac{1}{2}\log n})$.

A Simple Algorithm Based on FLT

For m different a's, test if $a^{n-1} = 1 \ (mod\ n)$.

- Takes $O(m \log n)$ arithmetic operations.
- However, it goes wrong on some numbers, for example, Carmichael numbers.
  - These are composite numbers with the property that for every $a < n$, $a^n = a \ (mod\ n)$.
  - There exist infinitely many Carmichael numbers with $561 = 3 * 11 * 17$ the smallest one.

A Simple Algorithm Based on FLT

For m different a's, test if $a^{n-1} = 1 \ (mod\ n)$.

- Takes $O(m \log n)$ arithmetic operations.
- However, it goes wrong on some numbers, for example, Carmichael numbers.
  - These are composite numbers with the property that for every $a < n$, $a^n = a \ (mod\ n)$.
  - There exist infinitely many Carmichael numbers with $561 = 3 * 11 * 17$ the smallest one.

A Simple Algorithm Based on FLT

For m different a's, test if $a^{n-1} = 1 \ (mod\ n)$.

- Takes $O(m \log n)$ arithmetic operations.
- However, it goes wrong on some numbers, for example, Carmichael numbers.
  - These are composite numbers with the property that for every $a < n$, $a^n = a \ (mod\ n)$.
  - There exist infinitely many Carmichael numbers with $561 = 3 * 11 * 17$ the smallest one.

Outline

1. Fermat's Little Theorem
2. Primality Testing
3. Solovay-Strassen Algorithm
4. Miller-Rabin Algorithm
5. AKS Algorithm

Fermat's Little Theorem and Quadratic Residues

Theorem (A Restatement of FLT)

If n is odd prime then for every a, $1 \leq a < n$, $a^{\frac{n-1}{2}} = \pm 1 \pmod{n}$.

Fact

When n is prime, $a^{\frac{n-1}{2}} = 1 \pmod{n}$ iff a is a quadratic residue in Z_n.

Therefore, if n is prime then

$$\left(\frac{a}{n}\right) = a^{\frac{n-1}{2}} \pmod{n}.$$

Fermat's Little Theorem and Quadratic Residues

Theorem (A Restatement of FLT)

If n is odd prime then for every a, $1 \leq a < n$, $a^{\frac{n-1}{2}} = \pm 1 \ (mod\ n)$.

Fact

When n is prime, $a^{\frac{n-1}{2}} = 1 \ (mod\ n)$ iff a is a quadratic residue in Z_n.

Therefore, if n is prime then

$$\left(\frac{a}{n}\right) = a^{\frac{n-1}{2}} \ (mod\ n).$$

Fermat's Little Theorem and Quadratic Residues

Theorem (A Restatement of FLT)

If n is odd prime then for every a, $1 \leq a < n$, $a^{\frac{n-1}{2}} = \pm 1 \ (mod\ n)$.

Fact

When n is prime, $a^{\frac{n-1}{2}} = 1 \ (mod\ n)$ iff a is a quadratic residue in Z_n.

Therefore, if n is prime then

$$\left(\frac{a}{n}\right) = a^{\frac{n-1}{2}} \ (mod\ n).$$

Legendre-Jacobi Symbol

- For prime $n \geq 3$, $\left(\frac{a}{n}\right) = 1$ if a is a quadratic residue modulo n, -1 if a is a non-residue.
- If $n = \prod_{i=1}^{k} p_i^{e_i}$ with p_i's distinct odd primes then

$$\left(\frac{a}{n}\right) = \prod_{i=1}^{k} \left(\frac{a}{p_i}\right)^{e_i}$$

- It satisfies the quadratic reciprocity law:

$$\left(\frac{a}{n}\right) \cdot \left(\frac{n}{a}\right) = (-1)^{\frac{(a-1)(n-1)}{4}}$$

  for $n \geq 3$.
-

$$\left(\frac{a+n}{n}\right) = \left(\frac{a}{n}\right)$$

- Using last two properties, $\left(\frac{a}{n}\right)$ can be computed for odd n in $O(\log n)$ arithmetic operations.

Legendre-Jacobi Symbol

- For prime $n \geq 3$, $\left(\frac{a}{n}\right) = 1$ if a is a quadratic residue modulo n, -1 if a is a non-residue.
- If $n = \prod_{i=1}^{k} p_i^{e_i}$ with p_i's distinct odd primes then

$$\left(\frac{a}{n}\right) = \prod_{i=1}^{k} \left(\frac{a}{p_i}\right)^{e_i}$$

- It satisfies the quadratic reciprocity law:

$$\left(\frac{a}{n}\right) \cdot \left(\frac{n}{a}\right) = (-1)^{\frac{(a-1)(n-1)}{4}}$$

  for $n \geq 3$.
- $$\left(\frac{a+n}{n}\right) = \left(\frac{a}{n}\right)$$
- Using last two properties, $\left(\frac{a}{n}\right)$ can be computed for odd n in $O(\log n)$ arithmetic operations.

Legendre-Jacobi Symbol

- For prime $n \geq 3$, $\left(\frac{a}{n}\right) = 1$ if a is a quadratic residue modulo n, -1 if a is a non-residue.
- If $n = \prod_{i=1}^{k} p_i^{e_i}$ with p_i's distinct odd primes then

$$\left(\frac{a}{n}\right) = \prod_{i=1}^{k} \left(\frac{a}{p_i}\right)^{e_i}$$

- It satisfies the quadratic reciprocity law:

$$\left(\frac{a}{n}\right) \cdot \left(\frac{n}{a}\right) = (-1)^{\frac{(a-1)(n-1)}{4}}$$

  for $n \geq 3$.

- $$\left(\frac{a+n}{n}\right) = \left(\frac{a}{n}\right)$$

- Using last two properties, $\left(\frac{a}{n}\right)$ can be computed for odd n in $O(\log n)$ arithmetic operations.

Legendre-Jacobi Symbol

- For prime $n \geq 3$, $\left(\frac{a}{n}\right) = 1$ if a is a quadratic residue modulo n, -1 if a is a non-residue.
- If $n = \prod_{i=1}^{k} p_i^{e_i}$ with p_i's distinct odd primes then

$$\left(\frac{a}{n}\right) = \prod_{i=1}^{k} \left(\frac{a}{p_i}\right)^{e_i}$$

- It satisfies the quadratic reciprocity law:

$$\left(\frac{a}{n}\right) \cdot \left(\frac{n}{a}\right) = (-1)^{\frac{(a-1)(n-1)}{4}}$$

  for $n \geq 3$.
- $$\left(\frac{a+n}{n}\right) = \left(\frac{a}{n}\right)$$

- Using last two properties, $\left(\frac{a}{n}\right)$ can be computed for odd n in $O(\log n)$ arithmetic operations.

Legendre-Jacobi Symbol

- For prime $n \geq 3$, $\left(\frac{a}{n}\right) = 1$ if a is a quadratic residue modulo n, -1 if a is a non-residue.
- If $n = \prod_{i=1}^{k} p_i^{e_i}$ with p_i's distinct odd primes then

$$\left(\frac{a}{n}\right) = \prod_{i=1}^{k} \left(\frac{a}{p_i}\right)^{e_i}$$

- It satisfies the quadratic reciprocity law:

$$\left(\frac{a}{n}\right) \cdot \left(\frac{n}{a}\right) = (-1)^{\frac{(a-1)(n-1)}{4}}$$

  for $n \geq 3$.
-

$$\left(\frac{a+n}{n}\right) = \left(\frac{a}{n}\right)$$

- Using last two properties, $\left(\frac{a}{n}\right)$ can be computed for odd n in $O(\log n)$ arithmetic operations.

Solovay-Strassen Algorithm

- Proposed by Solovay and Strassen (1973).
- A randomized algorithm based on the equation $\left(\frac{a}{n}\right) = a^{\frac{n-1}{2}}$ $(mod\ n)$.
- Never incorrectly classifies primes and correctly classifies composites with probability at least $\frac{1}{2}$.

Solovay-Strassen Algorithm

- Proposed by Solovay and Strassen (1973).
- A randomized algorithm based on the equation $\left(\frac{a}{n}\right) = a^{\frac{n-1}{2}} \ (mod\ n)$.
- Never incorrectly classifies primes and correctly classifies composites with probability at least $\frac{1}{2}$.

Solovay-Strassen Algorithm

- Proposed by Solovay and Strassen (1973).
- A randomized algorithm based on the equation $\left(\frac{a}{n}\right) = a^{\frac{n-1}{2}}\ (mod\ n)$.
- Never incorrectly classifies primes and correctly classifies composites with probability at least $\frac{1}{2}$.

Solovay-Strassen Algorithm

1. If $n = m^k$ for some $k > 1$ or an even number > 2, it is composite.
2. For a random a in Z_n, test if
$$\left(\frac{a}{n}\right) = a^{\frac{n-1}{2}} \ (mod\ n).$$
3. If yes, classify n as prime, otherwise as composite.

The time complexity is $O(\log n)$ arithmetic operations.

Solovay-Strassen Algorithm

1. If $n = m^k$ for some $k > 1$ or an even number > 2, it is composite.
2. For a random a in Z_n, test if

$$\left(\frac{a}{n}\right) = a^{\frac{n-1}{2}} \ (mod\ n).$$

3. If yes, classify n as prime, otherwise as composite.

The time complexity is $O(\log n)$ arithmetic operations.

Solovay-Strassen Algorithm

1. If $n = m^k$ for some $k > 1$ or an even number > 2, it is composite.
2. For a random a in Z_n, test if
$$\left(\frac{a}{n}\right) = a^{\frac{n-1}{2}} \ (mod\ n).$$
3. If yes, classify n as prime, otherwise as composite.

The time complexity is $O(\log n)$ arithmetic operations.

Solovay-Strassen Algorithm

1. If $n = m^k$ for some $k > 1$ or an even number > 2, it is composite.
2. For a random a in Z_n, test if
$$\left(\frac{a}{n}\right) = a^{\frac{n-1}{2}} \ (mod\ n).$$
3. If yes, classify n as prime, otherwise as composite.

The time complexity is $O(\log n)$ arithmetic operations.

Analysis

- If n is prime, it is always classified as prime.
- Consider the case when n is an odd composite and a product of at least two primes.
- Let $n = p^k \cdot m$ where p is prime, $k > 0$ is odd, and $(p, m) = 1$.

Facts

1. Every number $a < n$ can be uniquely decomposed as $a = \langle \alpha, c \rangle$ where $\alpha = a \ (mod\ p^k)$ and $c = a \ (mod\ m)$.
2. There are exactly $\frac{1}{2}(p-1)$ numbers between 0 and p that are quadratic residues modulo p.

Analysis

- If n is prime, it is always classified as prime.
- Consider the case when n is an odd composite and a product of at least two primes.
- Let $n = p^k \cdot m$ where p is prime, $k > 0$ is odd, and $(p, m) = 1$.

Facts

1. Every number $a < n$ can be uniquely decomposed as $a = \langle \alpha, c \rangle$ where $\alpha = a \ (mod\ p^k)$ and $c = a \ (mod\ m)$.
2. There are exactly $\frac{1}{2}(p-1)$ numbers between 0 and p that are quadratic residues modulo p.

Analysis

- If n is prime, it is always classified as prime.
- Consider the case when n is an odd composite and a product of at least two primes.
- Let $n = p^k \cdot m$ where p is prime, $k > 0$ is odd, and $(p, m) = 1$.

Facts

1. Every number $a < n$ can be uniquely decomposed as $a = \langle \alpha, c \rangle$ where $\alpha = a \ (mod\ p^k)$ and $c = a \ (mod\ m)$.
2. There are exactly $\frac{1}{2}(p-1)$ numbers between 0 and p that are quadratic residues modulo p.

Analysis

- Let $0 < \alpha, \beta < p$, $0 < c < m$ with α a quadratic residue modulo p and β a non-residue.
- Clearly,

$$\langle \alpha, c \rangle^{\frac{n-1}{2}} = \langle \beta, c \rangle^{\frac{n-1}{2}} = c^{\frac{n-1}{2}} \pmod{m}$$

- And

$$\left(\frac{\langle \alpha, c \rangle}{n}\right) = \left(\frac{\alpha}{p}\right)^k \cdot \left(\frac{c}{m}\right) = -\left(\frac{\beta}{p}\right)^k \cdot \left(\frac{c}{m}\right) = -\left(\frac{\langle \beta, c \rangle}{n}\right).$$

Analysis

- Let $0 < \alpha, \beta < p$, $0 < c < m$ with α a quadratic residue modulo p and β a non-residue.
- Clearly,

$$\langle \alpha, c \rangle^{\frac{n-1}{2}} = \langle \beta, c \rangle^{\frac{n-1}{2}} = c^{\frac{n-1}{2}} \ (mod\ m)$$

- And

$$\left(\frac{\langle \alpha, c \rangle}{n}\right) = \left(\frac{\alpha}{p}\right)^k \cdot \left(\frac{c}{m}\right) = -\left(\frac{\beta}{p}\right)^k \cdot \left(\frac{c}{m}\right) = -\left(\frac{\langle \beta, c \rangle}{n}\right).$$

Analysis

- Let $0 < \alpha, \beta < p$, $0 < c < m$ with α a quadratic residue modulo p and β a non-residue.
- Clearly,

$$\langle \alpha, c \rangle^{\frac{n-1}{2}} = \langle \beta, c \rangle^{\frac{n-1}{2}} = c^{\frac{n-1}{2}} \ (mod\ m)$$

- And

$$\left(\frac{\langle \alpha, c \rangle}{n}\right) = \left(\frac{\alpha}{p}\right)^k \cdot \left(\frac{c}{m}\right) = -\left(\frac{\beta}{p}\right)^k \cdot \left(\frac{c}{m}\right) = -\left(\frac{\langle \beta, c \rangle}{n}\right).$$

Analysis

- If $\langle \alpha, c\rangle^{\frac{n-1}{2}} \neq \langle \beta, c\rangle^{\frac{n-1}{2}} \ (mod\ n)$ then one of them is not in $\{1, -1\}$ and so compositeness of n is proven.
- Otherwise, either

$$\left(\frac{\langle \alpha, c\rangle}{n}\right) \neq \langle \alpha, c\rangle^{\frac{n-1}{2}} \ (mod\ n),$$

or

$$\left(\frac{\langle \beta, c\rangle}{n}\right) \neq \langle \beta, c\rangle^{\frac{n-1}{2}} \ (mod\ n).$$

Analysis

- If $\langle\alpha, c\rangle^{\frac{n-1}{2}} \neq \langle\beta, c\rangle^{\frac{n-1}{2}}$ $(mod\ n)$ then one of them is not in $\{1, -1\}$ and so compositeness of n is proven.
- Otherwise, either

$$\left(\frac{\langle\alpha, c\rangle}{n}\right) \neq \langle\alpha, c\rangle^{\frac{n-1}{2}}\ (mod\ n),$$

or

$$\left(\frac{\langle\beta, c\rangle}{n}\right) \neq \langle\beta, c\rangle^{\frac{n-1}{2}}\ (mod\ n).$$

Analysis

- So it cannot be that both a quadratic residue and a non-residue modulo p satisfy the equation

$$\left(\frac{a}{n}\right) = a^{\frac{n-1}{2}} \ (mod\ n).$$

- Therefore, with probability at least $\frac{1}{2}$, when n is composite, the algorithm will be correct.

Analysis

- So it cannot be that both a quadratic residue and a non-residue modulo p satisfy the equation

$$\left(\frac{a}{n}\right) = a^{\frac{n-1}{2}} \ (mod\ n).$$

- Therefore, with probability at least $\frac{1}{2}$, when n is composite, the algorithm will be correct.

Outline

1 Fermat's Little Theorem

2 Primality Testing

3 Solovay-Strassen Algorithm

4 Miller-Rabin Algorithm

5 AKS Algorithm

Fermat's Little Theorem and More Quadratic Residues

Theorem (Another Restatement of FLT)

If n is odd prime and $n = 1 + 2^s \cdot t$, t odd, then for every a, $1 \leq a < n$, the sequence $a^{\frac{n-1}{2}} = a^{2^{s-1} \cdot t} \pmod{n}$, $a^{2^{s-2} \cdot t} \pmod{n}$, …, $a^t \pmod{n}$ has either all 1's or a -1 somewhere.

Miller's Algorithm

- This theorem is the basis for Miller's algorithm (1973).
- It is a deterministic polynomial time test.
- It is correct under Extended Riemann Hypothesis.

Miller's Algorithm

- This theorem is the basis for Miller's algorithm (1973).
- It is a deterministic polynomial time test.
- It is correct under Extended Riemann Hypothesis.

Miller's Algorithm

- This theorem is the basis for Miller's algorithm (1973).
- It is a deterministic polynomial time test.
- It is correct under Extended Riemann Hypothesis.

Miller's Algorithm

1. If $n = m^k$ for some $k > 1$ or is even number > 2, it is composite.
2. For each a, $1 < a \leq 4\log^2 n$, check if the sequence $a^{2^{s-1}\cdot t}$ $(mod\ n)$, $a^{2^{s-2}\cdot t}$ $(mod\ n)$, ..., a^t $(mod\ n)$ has either all 1's or a -1 somewhere.
3. If yes, classify n as prime, otherwise as composite.

The time complexity of the test is $O(\log^3 n)$ arithmetic operations.

Miller's Algorithm

1. If $n = m^k$ for some $k > 1$ or is even number > 2, it is composite.
2. For each a, $1 < a \leq 4\log^2 n$, check if the sequence $a^{2^{s-1}\cdot t} \ (mod\ n)$, $a^{2^{s-2}\cdot t} \ (mod\ n)$, ..., $a^t \ (mod\ n)$ has either all 1's or a -1 somewhere.
3. If yes, classify n as prime, otherwise as composite.

The time complexity of the test is $O(\log^3 n)$ arithmetic operations.

Miller's Algorithm

1. If $n = m^k$ for some $k > 1$ or is even number > 2, it is composite.
2. For each a, $1 < a \leq 4\log^2 n$, check if the sequence $a^{2^{s-1}\cdot t}$ $(mod\ n)$, $a^{2^{s-2}\cdot t}$ $(mod\ n)$, ..., a^t $(mod\ n)$ has either all 1's or a -1 somewhere.
3. If yes, classify n as prime, otherwise as composite.

The time complexity of the test is $O(\log^3 n)$ arithmetic operations.

Miller's Algorithm

1. If $n = m^k$ for some $k > 1$ or is even number > 2, it is composite.
2. For each a, $1 < a \leq 4\log^2 n$, check if the sequence $a^{2^{s-1}\cdot t}$ $(mod\ n)$, $a^{2^{s-2}\cdot t}$ $(mod\ n)$, $\ldots$, a^t $(mod\ n)$ has either all 1's or a -1 somewhere.
3. If yes, classify n as prime, otherwise as composite.

The time complexity of the test is $O(\log^3 n)$ arithmetic operations.

Rabin's Modification

- A modification of Miller's algorithm proposed soon after (1974).
- Selects a randomly instead of trying all a in the range $[2, 4\log^2 n]$.
- Randomized algorithm that never classfies primes incorrectly and correctly classifies composites with probabilty at least $\frac{3}{4}$.
- Time complexity is $O(\log n)$ arithmetic operations.
- The most popular primality testing algorithm.

Rabin's Modification

- A modification of Miller's algorithm proposed soon after (1974).
- Selects a randomly instead of trying all a in the range $[2, 4\log^2 n]$.
- Randomized algorithm that never classfies primes incorrectly and correctly classifies composites with probabilty at least $\frac{3}{4}$.
- Time complexity is $O(\log n)$ arithmetic operations.
- The most popular primality testing algorithm.

Rabin's Modification

- A modification of Miller's algorithm proposed soon after (1974).
- Selects a randomly instead of trying all a in the range $[2, 4\log^2 n]$.
- Randomized algorithm that never classfies primes incorrectly and correctly classifies composites with probabilty at least $\frac{3}{4}$.
- Time complexity is $O(\log n)$ arithmetic operations.
- The most popular primality testing algorithm.

Rabin's Modification

- A modification of Miller's algorithm proposed soon after (1974).
- Selects a randomly instead of trying all a in the range $[2, 4\log^2 n]$.
- Randomized algorithm that never classfies primes incorrectly and correctly classifies composites with probabilty at least $\frac{3}{4}$.
- Time complexity is $O(\log n)$ arithmetic operations.
- The most popular primality testing algorithm.

Outline

1 Fermat's Little Theorem

2 Primality Testing

3 Solovay-Strassen Algorithm

4 Miller-Rabin Algorithm

5 AKS Algorithm

Fermat's Little Theorem for Polynomials

Theorem (A Third Generalization of FLT)

If n is prime then for every a, $1 \leq a < n$,
$(x + a)^n = x^n + a \pmod{n, x^r - 1}$.

$P(x) \pmod{n, x^r - 1}$ is the residue polynomial obtained by reducing its coefficients modulo n and powers of x modulo r.

Proof

- We have
$$(x+a)^n = \sum_{i=0}^{n} \binom{n}{i} x^i \cdot a^{n-i}.$$
- Since n is prime, each of $\binom{n}{i}$ is divisible by n for $1 \le i < n$.
- Also, $a^n = a \ (mod\ n)$.
- Therefore,
$$(x+a)^n = x^n + a \ (mod\ n).$$

Proof

- We have

$$(x+a)^n = \sum_{i=0}^{n} \binom{n}{i} x^i \cdot a^{n-i}.$$

- Since n is prime, each of $\binom{n}{i}$ is divisible by n for $1 \leq i < n$.
- Also, $a^n = a \ (mod\ n)$.
- Therefore,

$$(x+a)^n = x^n + a \ (mod\ n).$$

Proof

- We have
$$(x+a)^n = \sum_{i=0}^{n} \binom{n}{i} x^i \cdot a^{n-i}.$$
- Since n is prime, each of $\binom{n}{i}$ is divisible by n for $1 \leq i < n$.
- Also, $a^n = a \ (mod\ n)$.
- Therefore,
$$(x+a)^n = x^n + a \ (mod\ n).$$

AKS Algorithm

- A test proposed in 2002 based on the above generalization.
- The time complexity is $O(\log^{\frac{19}{2}} n)$ arithmetic operations.
- The only known deterministic, unconditionally correct, polynomial time algorithm.

AKS Algorithm

- A test proposed in 2002 based on the above generalization.
- The time complexity is $O(\log^{\frac{19}{2}} n)$ arithmetic operations.
- The only known deterministic, unconditionally correct, polynomial time algorithm.

AKS Algorithm

- A test proposed in 2002 based on the above generalization.
- The time complexity is $O(\log^{\frac{19}{2}} n)$ arithmetic operations.
- The only known deterministic, unconditionally correct, polynomial time algorithm.

AKS Algorithm

1. If $n = m^k$ for $k > 1$, or is even or has a small divisor, it is composite.
2. Find the smallest number r such that $O_r(n) > 4\log^2 n$.
3. For each a, $1 \leq a \leq 2\sqrt{r}\log n$, check if $(x+a)^n = x^n + a \ (mod \ n, x^r - 1)$.
4. If yes, n is prime, otherwise composite.

$O_r(n)$ is the smallest number k for which $n^k = 1 \ (mod \ r)$.

AKS Algorithm

1. If $n = m^k$ for $k > 1$, or is even or has a small divisor, it is composite.
2. Find the smallest number r such that $O_r(n) > 4\log^2 n$.
3. For each a, $1 \leq a \leq 2\sqrt{r}\log n$, check if $(x+a)^n = x^n + a\ (mod\ n, x^r - 1)$.
4. If yes, n is prime, otherwise composite.

$O_r(n)$ is the smallest number k for which $n^k = 1\ (mod\ r)$.

AKS Algorithm

1. If $n = m^k$ for $k > 1$, or is even or has a small divisor, it is composite.
2. Find the smallest number r such that $O_r(n) > 4\log^2 n$.
3. For each a, $1 \leq a \leq 2\sqrt{r}\log n$, check if $(x+a)^n = x^n + a \ (mod\ n, x^r - 1)$.
4. If yes, n is prime, otherwise composite.

$O_r(n)$ is the smallest number k for which $n^k = 1 \ (mod\ r)$.

AKS Algorithm

1. If $n = m^k$ for $k > 1$, or is even or has a small divisor, it is composite.
2. Find the smallest number r such that $O_r(n) > 4\log^2 n$.
3. For each a, $1 \leq a \leq 2\sqrt{r}\log n$, check if $(x+a)^n = x^n + a \ (mod\ n, x^r - 1)$.
4. If yes, n is prime, otherwise composite.

$O_r(n)$ is the smallest number k for which $n^k = 1 \ (mod\ r)$.

Analysis: Two Sets

- Suppose n has a prime factor p.
- Fix r such that $O_r(n) > 4\log^2 n$.
- Suppose that for each $1 \leq a \leq 2\sqrt{r}\log n$,

$$(x+a)^n = x^n + a \ (mod\ n, x^r - 1).$$

- Define two sets A and B as follows:

$$\begin{aligned} A &= \{m \mid (x+a)^m = x^m + a \ (mod\ p, x^r - 1) \\ &\qquad \text{for every } a,\ 1 \leq a \leq 2\sqrt{r}\log n\} \\ B &= \{g(x) \mid g(x)^m = g(x^m) \ (mod\ p, x^r - 1) \text{ for every } m \in A\} \end{aligned}$$

- We have $n, p \in A$ and $x + a \in B$ for $1 \leq a \leq 2\sqrt{r}\log n$.

Analysis: Two Sets

- Suppose n has a prime factor p.
- Fix r such that $O_r(n) > 4 \log^2 n$.
- Suppose that for each $1 \leq a \leq 2\sqrt{r} \log n$,

$$(x+a)^n = x^n + a \ (mod\ n, x^r - 1).$$

- Define two sets A and B as follows:

$$A = \{m \mid (x+a)^m = x^m + a \ (mod\ p, x^r - 1)$$
$$\text{for every } a,\ 1 \leq a \leq 2\sqrt{r} \log n\}$$
$$B = \{g(x) \mid g(x)^m = g(x^m) \ (mod\ p, x^r - 1) \text{ for every } m \in A\}$$

- We have $n, p \in A$ and $x + a \in B$ for $1 \leq a \leq 2\sqrt{r} \log n$.

Analysis: Two Sets

- Suppose n has a prime factor p.
- Fix r such that $O_r(n) > 4\log^2 n$.
- Suppose that for each $1 \le a \le 2\sqrt{r}\log n$,

$$(x+a)^n = x^n + a \; (mod \; n, x^r - 1).$$

- Define two sets A and B as follows:

$$\begin{aligned} A &= \{m \mid (x+a)^m = x^m + a \; (mod \; p, x^r - 1) \\ &\qquad\qquad \text{for every } a,\ 1 \le a \le 2\sqrt{r}\log n\} \\ B &= \{g(x) \mid g(x)^m = g(x^m) \; (mod \; p, x^r - 1) \text{ for every } m \in A\} \end{aligned}$$

- We have $n, p \in A$ and $x + a \in B$ for $1 \le a \le 2\sqrt{r}\log n$.

Analysis: Two Sets

- Suppose n has a prime factor p.
- Fix r such that $O_r(n) > 4\log^2 n$.
- Suppose that for each $1 \le a \le 2\sqrt{r}\log n$,

$$(x+a)^n = x^n + a \; (mod \; n, x^r - 1).$$

- Define two sets A and B as follows:

$$\begin{aligned} A &= \{m \mid (x+a)^m = x^m + a \; (mod \; p, x^r - 1) \\ &\qquad\qquad \text{for every } a,\; 1 \le a \le 2\sqrt{r}\log n\} \\ B &= \{g(x) \mid g(x)^m = g(x^m) \; (mod \; p, x^r - 1) \text{ for every } m \in A\} \end{aligned}$$

- We have $n, p \in A$ and $x + a \in B$ for $1 \le a \le 2\sqrt{r}\log n$.

Analysis: Two More Sets

Observation

Both A and B are closed under multiplication.

- Now define two more sets:

$$
\begin{aligned}
A_0 &= \{m \pmod{r} \mid m \in A\} \\
B_0 &= \{g(x) \pmod{p, h(x)} \mid g(x) \in B\}
\end{aligned}
$$

  Here $h(x)$ is an irreducible factor of $x^r - 1$ modulo p containing a primitive rth root of unity.
- Let $F = F_p[x]/(h(x))$, the field of polynomials modulo p and $h(x)$.

Analysis: Two More Sets

Observation

Both A and B are closed under multiplication.

- Now define two more sets:

$$\begin{aligned} A_0 &= \{m \ (mod\ r) \mid m \in A\} \\ B_0 &= \{g(x) \ (mod\ p, h(x)) \mid g(x) \in B\} \end{aligned}$$

  Here $h(x)$ is an irreducible factor of $x^r - 1$ modulo p containing a primitive rth root of unity.

- Let $F = F_p[x]/(h(x))$, the field of polynomials modulo p and $h(x)$.

Analysis: Two More Sets

Observation

Both A and B are closed under multiplication.

- Now define two more sets:

$$\begin{aligned} A_0 &= \{m \ (mod\ r) \mid m \in A\} \\ B_0 &= \{g(x) \ (mod\ p, h(x)) \mid g(x) \in B\} \end{aligned}$$

  Here $h(x)$ is an irreducible factor of $x^r - 1$ modulo p containing a primitive rth root of unity.

- Let $F = F_p[x]/(h(x))$, the field of polynomials modulo p and $h(x)$.

Analysis: Estimating Size of A_0

Recall

$A_0 = \{m \pmod r \mid m \in A\}$

- Let $t = |A_0|$.
- Since elements of A_0 are non-zero numbers modulo r, $t < r$.
- Since all powers of n belong to A and $O_r(n) > 4\log^2 n$, $t > 4\log^2 n$.
- Hence,

$$4\log^2 n < t < r.$$

Analysis: Estimating Size of A_0

Recall

$A_0 = \{m \pmod r \mid m \in A\}$

- Let $t = |A_0|$.
- Since elements of A_0 are non-zero numbers modulo r, $t < r$.
- Since all powers of n belong to A and $O_r(n) > 4\log^2 n$, $t > 4\log^2 n$.
- Hence,

$$4\log^2 n < t < r.$$

Analysis: Estimating Size of A_0

Recall

$A_0 = \{m \ (mod\ r) \mid m \in A\}$

- Let $t = |A_0|$.
- Since elements of A_0 are non-zero numbers modulo r, $t < r$.
- Since all powers of n belong to A and $O_r(n) > 4\log^2 n$, $t > 4\log^2 n$.
- Hence,
$$4\log^2 n < t < r.$$

Analysis: Estimating Size of A_0

Recall

$A_0 = \{m \ (mod\ r) \mid m \in A\}$

- Let $t = |A_0|$.
- Since elements of A_0 are non-zero numbers modulo r, $t < r$.
- Since all powers of n belong to A and $O_r(n) > 4\log^2 n$, $t > 4\log^2 n$.
- Hence,

$$4\log^2 n < t < r.$$

Analysis: Estimating Size of B_0

Recall

$B_0 = \{g(x) \ (mod \ p, h(x)) \mid g(x) \in B\}$

- Let $T = |B_0|$.
- Since elements of B_0 are polynomials modulo p and $h(x)$ and degree of h is $< r$, $T \leq p^{r-1}$.
- The lower bound on T is trickier.

Analysis: Estimating Size of B_0

Recall

$B_0 = \{g(x) \pmod{p, h(x)} \mid g(x) \in B\}$

- Let $T = |B_0|$.
- Since elements of B_0 are polynomials modulo p and $h(x)$ and degree of h is $< r$, $T \leq p^{r-1}$.
- The lower bound on T is trickier.

Analysis: Estimating Size of B_0

Recall

$B_0 = \{g(x) \ (mod \ p, h(x)) \mid g(x) \in B\}$

- Let $T = |B_0|$.
- Since elements of B_0 are polynomials modulo p and $h(x)$ and degree of h is $< r$, $T \leq p^{r-1}$.
- The lower bound on T is trickier.

Analysis: Lower Bound on T

Recall

$B_0 = \{g(x) \ (mod\ p, h(x)) \mid g(x) \in B\}$

- Consider $f(x), g(x) \in B$, $f \neq g$ and both of degree $< t$.
- Suppose $f(x) = g(x)$ in B_0, i.e., $f(x) = g(x) \ (mod\ p, h(x))$.
- Since $f, g \in B$, $f(x)^m = f(x^m) \ (mod\ p, x^r - 1)$ and $g(x)^m = g(x^m) \ (mod\ p, x^r - 1)$ for every $m \in A$.
- Therefore, $f(x^m) = g(x^m) \ (mod\ p, h(x))$ for every $m \in A_0$.
- Since x is a primitive rth root of unity in F, we get $|A_0| = t$ distinct roots of the polynomial $f(y) - g(y)$ in F.
- Not possible since degree of $f(y) - g(y)$ is $< t$ and F is a field.

Analysis: Lower Bound on T

Recall

$B_0 = \{g(x) \ (mod\ p, h(x)) \mid g(x) \in B\}$

- Consider $f(x), g(x) \in B$, $f \neq g$ and both of degree $< t$.
- Suppose $f(x) = g(x)$ in B_0, i.e., $f(x) = g(x) \ (mod\ p, h(x))$.
- Since $f, g \in B$, $f(x)^m = f(x^m) \ (mod\ p, x^r - 1)$ and $g(x)^m = g(x^m) \ (mod\ p, x^r - 1)$ for every $m \in A$.
- Therefore, $f(x^m) = g(x^m) \ (mod\ p, h(x))$ for every $m \in A_0$.
- Since x is a primitive rth root of unity in F, we get $|A_0| = t$ distinct roots of the polynomial $f(y) - g(y)$ in F.
- Not possible since degree of $f(y) - g(y)$ is $< t$ and F is a field.

Analysis: Lower Bound on T

Recall

$B_0 = \{g(x)\ (mod\ p, h(x)) \mid g(x) \in B\}$

- Consider $f(x), g(x) \in B$, $f \neq g$ and both of degree $< t$.
- Suppose $f(x) = g(x)$ in B_0, i.e., $f(x) = g(x)\ (mod\ p, h(x))$.
- Since $f, g \in B$, $f(x)^m = f(x^m)\ (mod\ p, x^r - 1)$ and $g(x)^m = g(x^m)\ (mod\ p, x^r - 1)$ for every $m \in A$.
- Therefore, $f(x^m) = g(x^m)\ (mod\ p, h(x))$ for every $m \in A_0$.
- Since x is a primitive rth root of unity in F, we get $|A_0| = t$ distinct roots of the polynomial $f(y) - g(y)$ in F.
- Not possible since degree of $f(y) - g(y)$ is $< t$ and F is a field.

Analysis: Lower Bound on T

Recall

$B_0 = \{g(x) \ (mod\ p, h(x)) \mid g(x) \in B\}$

- Consider $f(x), g(x) \in B$, $f \neq g$ and both of degree $< t$.
- Suppose $f(x) = g(x)$ in B_0, i.e., $f(x) = g(x) \ (mod\ p, h(x))$.
- Since $f, g \in B$, $f(x)^m = f(x^m) \ (mod\ p, x^r - 1)$ and $g(x)^m = g(x^m) \ (mod\ p, x^r - 1)$ for every $m \in A$.
- Therefore, $f(x^m) = g(x^m) \ (mod\ p, h(x))$ for every $m \in A_0$.
- Since x is a primitive rth root of unity in F, we get $|A_0| = t$ distinct roots of the polynomial $f(y) - g(y)$ in F.
- Not possible since degree of $f(y) - g(y)$ is $< t$ and F is a field.

Analysis: Lower Bound on T

Recall

$B_0 = \{g(x) \ (mod\ p, h(x)) \mid g(x) \in B\}$

- Consider $f(x), g(x) \in B$, $f \neq g$ and both of degree $< t$.
- Suppose $f(x) = g(x)$ in B_0, i.e., $f(x) = g(x) \ (mod\ p, h(x))$.
- Since $f, g \in B$, $f(x)^m = f(x^m) \ (mod\ p, x^r - 1)$ and $g(x)^m = g(x^m) \ (mod\ p, x^r - 1)$ for every $m \in A$.
- Therefore, $f(x^m) = g(x^m) \ (mod\ p, h(x))$ for every $m \in A_0$.
- Since x is a primitive rth root of unity in F, we get $|A_0| = t$ distinct roots of the polynomial $f(y) - g(y)$ in F.
- Not possible since degree of $f(y) - g(y)$ is $< t$ and F is a field.

Analysis: Lower Bound on T

Recall

$B_0 = \{g(x)\ (mod\ p, h(x)) \mid g(x) \in B\}$

- Consider $f(x), g(x) \in B$, $f \neq g$ and both of degree $< t$.
- Suppose $f(x) = g(x)$ in B_0, i.e., $f(x) = g(x)\ (mod\ p, h(x))$.
- Since $f, g \in B$, $f(x)^m = f(x^m)\ (mod\ p, x^r - 1)$ and $g(x)^m = g(x^m)\ (mod\ p, x^r - 1)$ for every $m \in A$.
- Therefore, $f(x^m) = g(x^m)\ (mod\ p, h(x))$ for every $m \in A_0$.
- Since x is a primitive rth root of unity in F, we get $|A_0| = t$ distinct roots of the polynomial $f(y) - g(y)$ in F.
- Not possible since degree of $f(y) - g(y)$ is $< t$ and F is a field.

Analysis: Lower Bound on T

Recall

$B_0 = \{g(x) \ (mod\ p, h(x)) \mid g(x) \in B\}$

- We have that every distinct polynomial of degree $< t$ in B is a distinct element in B_0.
- B has at least $2\sqrt{r}\log n$ polynomials of degree 1 and is closed under multiplication.
- The number of distinct polynomials of degree $< t$ in B is more than $n^{2\sqrt{t}}$.
- Therefore,

$$n^{2\sqrt{t}} < T \leq p^{r-1}.$$

Analysis: Lower Bound on T

Recall

$B_0 = \{g(x) \ (mod\ p, h(x)) \mid g(x) \in B\}$

- We have that every distinct polynomial of degree $< t$ in B is a distinct element in B_0.
- B has at least $2\sqrt{r}\log n$ polynomials of degree 1 and is closed under multiplication.
- The number of distinct polynomials of degree $< t$ in B is more than $n^{2\sqrt{t}}$.
- Therefore,

$$n^{2\sqrt{t}} < T \leq p^{r-1}.$$

Analysis: Lower Bound on T

Recall

$B_0 = \{g(x) \ (mod\ p, h(x)) \mid g(x) \in B\}$

- We have that every distinct polynomial of degree $< t$ in B is a distinct element in B_0.
- B has at least $2\sqrt{r}\log n$ polynomials of degree 1 and is closed under multiplication.
- The number of distinct polynomials of degree $< t$ in B is more than $n^{2\sqrt{t}}$.
- Therefore,

$$n^{2\sqrt{t}} < T \leq p^{r-1}.$$

Analysis: Lower Bound on T

Recall

$B_0 = \{g(x) \ (mod\ p, h(x)) \mid g(x) \in B\}$

- We have that every distinct polynomial of degree $< t$ in B is a distinct element in B_0.
- B has at least $2\sqrt{r}\log n$ polynomials of degree 1 and is closed under multiplication.
- The number of distinct polynomials of degree $< t$ in B is more than $n^{2\sqrt{t}}$.
- Therefore,

$$n^{2\sqrt{t}} < T \leq p^{r-1}.$$

Analysis: Relation Between n and p

- Since $|A_0| = t$, there exist two pairs $(i,j) \neq (k,\ell)$, $i,j,k,\ell \leq \sqrt{t}$ such that
$$n^i \cdot p^j = n^k \cdot p^\ell \ (mod\ r).$$
- Let $g(x) \in B_0$.
- We have:
$$g(x)^{n^i \cdot p^j} = g(x^{n^i \cdot p^j}) = g(x^{n^k \cdot p^\ell}) = g(x)^{n^k \cdot p^\ell} \ (mod\ p, h(x)).$$
- Therefore, every $g(x) \in B_0$ is a root of the polynomial $y^{n^i \cdot p^j} - y^{n^k \cdot p^\ell}$.

Analysis: Relation Between n and p

- Since $|A_0| = t$, there exist two pairs $(i,j) \neq (k,\ell)$, $i,j,k,\ell \leq \sqrt{t}$ such that
$$n^i \cdot p^j = n^k \cdot p^\ell \ (mod\ r).$$

- Let $g(x) \in B_0$.
- We have:
$$g(x)^{n^i \cdot p^j} = g(x^{n^i \cdot p^j}) = g(x^{n^k \cdot p^\ell}) = g(x)^{n^k \cdot p^\ell} \ (mod\ p, h(x)).$$

- Therefore, every $g(x) \in B_0$ is a root of the polynomial $y^{n^i \cdot p^j} - y^{n^k \cdot p^\ell}$.

Analysis: Relation Between n and p

- Since $|A_0| = t$, there exist two pairs $(i, j) \neq (k, \ell)$, $i, j, k, \ell \leq \sqrt{t}$ such that
$$n^i \cdot p^j = n^k \cdot p^\ell \ (mod\ r).$$
- Let $g(x) \in B_0$.
- We have:
$$g(x)^{n^i \cdot p^j} = g(x^{n^i \cdot p^j}) = g(x^{n^k \cdot p^\ell}) = g(x)^{n^k \cdot p^\ell} \ (mod\ p, h(x)).$$
- Therefore, every $g(x) \in B_0$ is a root of the polynomial $y^{n^i \cdot p^j} - y^{n^k \cdot p^\ell}$.

Analysis: Relation Between n and p

- By the choice, $n^i \cdot p^j,\ n^k \cdot p^\ell \leq n^{2\sqrt{t}}$.
- Since $T > n^{2\sqrt{t}}$, the polynomial $y^{n^i \cdot p^j} - y^{n^k \cdot p^\ell}$ must be zero.
- This implies, $n^i \cdot p^j = n^k \cdot p^\ell$.
- Since p is a prime, p divides n, and $(i, j) \neq (k, \ell)$, we get

  $$n = p^c$$

  for some $c > 0$.
- Therefore, n must be prime.

Analysis: Relation Between n and p

- By the choice, $n^i \cdot p^j$, $n^k \cdot p^\ell \leq n^{2\sqrt{t}}$.
- Since $T > n^{2\sqrt{t}}$, the polynomial $y^{n^i \cdot p^j} - y^{n^k \cdot p^\ell}$ must be zero.
- This implies, $n^i \cdot p^j = n^k \cdot p^\ell$.
- Since p is a prime, p divides n, and $(i, j) \neq (k, \ell)$, we get

$$n = p^c$$

  for some $c > 0$.
- Therefore, n must be prime.

Analysis: Relation Between n and p

- By the choice, $n^i \cdot p^j$, $n^k \cdot p^\ell \leq n^{2\sqrt{t}}$.
- Since $T > n^{2\sqrt{t}}$, the polynomial $y^{n^i \cdot p^j} - y^{n^k \cdot p^\ell}$ must be zero.
- This implies, $n^i \cdot p^j = n^k \cdot p^\ell$.
- Since p is a prime, p divides n, and $(i, j) \neq (k, \ell)$, we get

$$n = p^c$$

  for some $c > 0$.
- Therefore, n must be prime.

Analysis: Relation Between n and p

- By the choice, $n^i \cdot p^j$, $n^k \cdot p^\ell \leq n^{2\sqrt{t}}$.
- Since $T > n^{2\sqrt{t}}$, the polynomial $y^{n^i \cdot p^j} - y^{n^k \cdot p^\ell}$ must be zero.
- This implies, $n^i \cdot p^j = n^k \cdot p^\ell$.
- Since p is a prime, p divides n, and $(i, j) \neq (k, \ell)$, we get

  $$n = p^c$$

  for some $c > 0$.
- Therefore, n must be prime.

Analysis: Relation Between n and p

- By the choice, $n^i \cdot p^j$, $n^k \cdot p^\ell \leq n^{2\sqrt{t}}$.
- Since $T > n^{2\sqrt{t}}$, the polynomial $y^{n^i \cdot p^j} - y^{n^k \cdot p^\ell}$ must be zero.
- This implies, $n^i \cdot p^j = n^k \cdot p^\ell$.
- Since p is a prime, p divides n, and $(i, j) \neq (k, \ell)$, we get

$$n = p^c$$

  for some $c > 0$.
- Therefore, n must be prime.

Analysis: Estimating Size of r

- Number r is such that $O_r(n) > 4\log^2 n$.
- It can be any prime not dividing the product

$$\prod_{i=1}^{4\log^2 n} (n^i - 1) < n^{16\log^4 n}.$$

- It follows that $r = O(\log^5 n)$.

Analysis: Estimating Size of r

- Number r is such that $O_r(n) > 4\log^2 n$.
- It can be any prime not dividing the product

$$\prod_{i=1}^{4\log^2 n} (n^i - 1) < n^{16\log^4 n}.$$

- It follows that $r = O(\log^5 n)$.

Analysis: Estimating Size of r

- Number r is such that $O_r(n) > 4\log^2 n$.
- It can be any prime not dividing the product

$$\prod_{i=1}^{4\log^2 n} (n^i - 1) < n^{16\log^4 n}.$$

- It follows that $r = O(\log^5 n)$.

Time Complexity Improvements

- Lenstra and Pomerance (2003) further reduce the size of r to $O(\log^2 n)$ resulting in the time complexity of $O(\log^5 n)$ arithmetic operations.
- Bernstein (2003) reduced the time complexity to $O(\log^3 n)$ arithmetic operations at the cost of making it randomized.

Time Complexity Improvements

- Lenstra and Pomerance (2003) further reduce the size of r to $O(\log^2 n)$ resulting in the time complexity of $O(\log^5 n)$ arithmetic operations.
- Bernstein (2003) reduced the time complexity to $O(\log^3 n)$ arithmetic operations at the cost of making it randomized.