Introduction to Algorithms
Lecture 13 Number Theoretic Algorithm

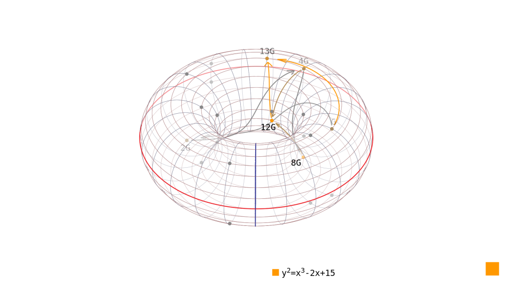
Xue Chen
xuechen1989@ustc.edu.cn
2025 spring in

Outline

Introduction

Many algorithms uses number theory

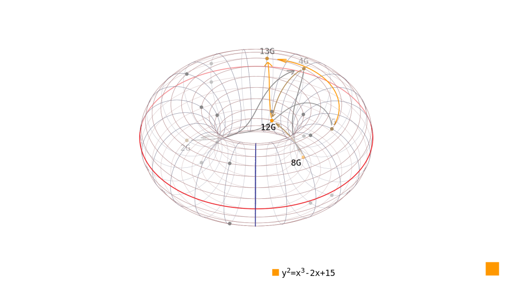
1. Hash functions
2. Coding theory
3. Cryptography

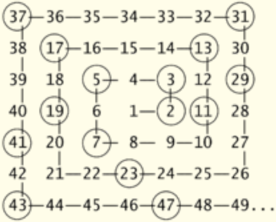


Introduction

Many algorithms uses number theory

1. Hash functions
2. Coding theory
3. Cryptography





Primes are the backbone of Number theory

Most Fundamental Problem in Number Theory

How to find a large prime number p?

Set up

Our Focus

Given n say 10^3 or 10^4, find a prime number of n digits (in binary).

Set up

Our Focus

Given n say 10^3 or 10^4, find a prime number of n digits (in binary).

1. Basic idea: Sieve algorithm?

Set up

Our Focus

Given n say 10^3 or 10^4, find a prime number of n digits (in binary).

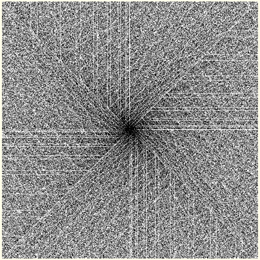
1. Basic idea: Sieve algorithm?
2. But we want a number between 2^{n-1} and $2^n - 1$!

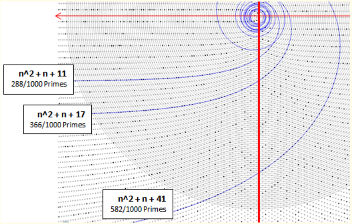
Set up

Our Focus

Given n say 10^3 or 10^4, find a prime number of n digits (in binary).

1. Basic idea: Sieve algorithm?
2. But we want a number between 2^{n-1} and $2^n - 1$!
3. Lots of interesting math: prime number THM, Euclid's ALGO, Fermat's THM, chinese remainder THM, primality testing





Basic Algorithm

```
procedure GENERATE-PRIME(n)
    repeat
        Generate a random number q of n digits
    until q is a prime
```

Basic Algorithm

---

procedure GENERATE-PRIME(n)
  repeat
    Generate a random number q of n digits
  until q is a prime

To finish this algorithm and analyze it, lots of questions:

1. Density of primes in n-digit numbers?

Basic Algorithm

---

procedure GENERATE-PRIME(n)
  repeat
    Generate a random number q of n digits
  until q is a prime

To finish this algorithm and analyze it, lots of questions:

1. Density of primes in n-digit numbers?
2. How to test whether q is a prime or not efficiently?
3. Many tools ...

Basic Algorithm

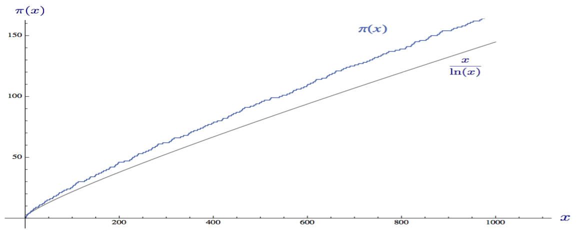
---

procedure GENERATE-PRIME(n)
  repeat
    Generate a random number q of n digits
  until q is a prime

To finish this algorithm and analyze it, lots of questions:

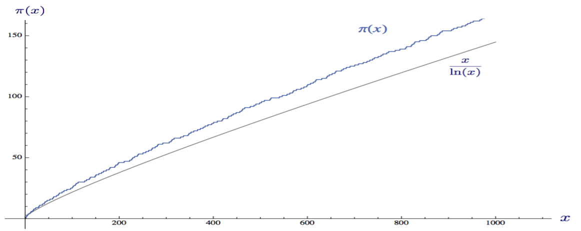
1. Density of primes in n-digit numbers?
2. How to test whether q is a prime or not efficiently?
3. Many tools ...
4. To the best of my knowledge, fastest algorithm in time $\approx n^3$

Outline



Prime Number Theorem — Most fundamental theorem of primes

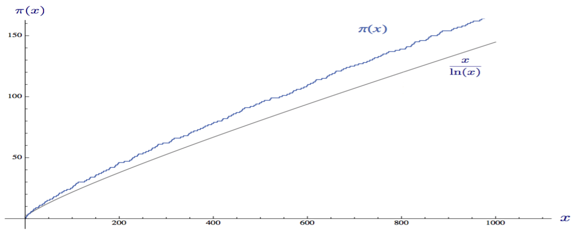
$\pi(X)$ denotes # primes less than X — $\Pi(N) \approx \frac{N}{\ln N}$.



Prime Number Theorem — Most fundamental theorem of primes

$\pi(X)$ denotes # primes less than X — $\Pi(N) \approx \frac{N}{\ln N}$.


1. Legendre, Gauss and others discovered the curve around 1800s, the THM was proved around 1890 by Hadamard and Poussin

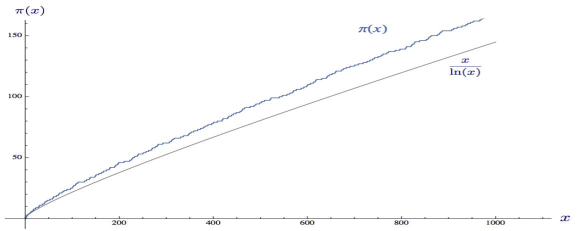


Prime Number Theorem — Most fundamental theorem of primes

$\pi(X)$ denotes # primes less than X — $\Pi(N) \approx \frac{N}{\ln N}$.



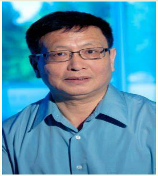
1. Legendre, Gauss and others discovered the curve around 1800s, the THM was proved around 1890 by Hadamard and Poussin
2. Riemann made a bolder claim 1852 — called Riemann hypothesis



Prime Number Theorem — Most fundamental theorem of primes

$\pi(X)$ denotes # primes less than X — $\Pi(N) \approx \frac{N}{\ln N}$.



1. Legendre, Gauss and others discovered the curve around 1800s, the THM was proved around 1890 by Hadamard and Poussin
2. Riemann made a bolder claim 1852 — called Riemann hypothesis
3. Twin prime number THM by Yitang Zhang 2013

Legendre

Weak Proof by Chebyshev

Let us show $\pi(N) = \Theta(N/\ln N)$

Weak Proof by Chebyshev

Let us show $\pi(N) = \Theta(N / \ln N)$

1. Consider $\binom{2N}{N}$ — all primes in $[N+1, 2N]$ are its factors

Weak Proof by Chebyshev

Let us show $\pi(N) = \Theta(N/\ln N)$

1. Consider $\binom{2N}{N}$ — all primes in $[N+1, 2N]$ are its factors
2. On the 1st hand, $\binom{2N}{N} = \Theta(2^{2N}/\sqrt{N})$

Weak Proof by Chebyshev

Let us show $\pi(N) = \Theta(N/\ln N)$

1. Consider $\binom{2N}{N}$ — all primes in $[N+1, 2N]$ are its factors
2. On the 1st hand, $\binom{2N}{N} = \Theta(2^{2N}/\sqrt{N})$
3. On the 2nd hand, let us consider the contribution of $p \leqslant N$ in $\binom{2N}{N}$
   — Question: Can we bound k s.t. at most k factors of p in $\binom{2N}{N}$?

Outline

Tools from number theory

Algorithms test whether a large integer is a prime or not — called primality testing. Before designing testers, what properties do prime number have?

Tools from number theory

Algorithms test whether a large integer is a prime or not — called primality testing. Before designing testers, what properties do prime number have?

1. Fix a large prime p of n digits and consider all operations in $\mod p$

Tools from number theory

Algorithms test whether a large integer is a prime or not — called primality testing. Before designing testers, what properties do prime number have?

1. Fix a large prime p of n digits and consider all operations in $\bmod p$
2. In group theory, all elements are in $Z_p = \{0, 1, \ldots, p-1\}$
3. Let $Z_p^* = \{1, \ldots, p-1\}$

Tools from number theory

Algorithms test whether a large integer is a prime or not — called primality testing. Before designing testers, what properties do prime number have?

1. Fix a large prime p of n digits and consider all operations in $\bmod p$
2. In group theory, all elements are in $Z_p = \{0, 1, \ldots, p-1\}$
3. Let $Z_p^* = \{1, \ldots, p-1\}$

Basic Property

For any $a \in \{1, \ldots, p-1\}$, $\exists b$ such that $ab \equiv 1 \mod p$, called a^{-1}.

In fact, we could find a^{-1} in time $O(\log p) = O(n)$ ☺

Revisit Euclid's Algorithm

Recall that Euclid's algorithm computes $gcd(a, b)$

Algorithm Euclid's algorithm for GCD

```
function EUCLID(a, b)
    if b=0 then
        return a
    else
        return Euclid(b, a mod b)
```

Revisit Euclid's Algorithm

Recall that Euclid's algorithm computes $gcd(a, b)$

Algorithm Extended Euclid's algorithm

function EUCLID(a, b)
  if b=0 **then**
    return $(x = 1, y = 0)$
  else
    $(x_0, y_0) = \text{Euclid}(b, a \mod b)$
    Return $(y_0, x_0 - [a/b] \cdot y_0)$

Extension

For any a, b, it finds $x, y \in \mathbb{Z}$ such that $ax + by = gcd(a, b)$

Properties of Prime

Fermat's little Theorem

Fix a prime p, for any $a \in Z_p^*$, $a^{p-1} \equiv 1 \mod p$.

1. Proof?

Properties of Prime

Fermat's little Theorem

Fix a prime p, for any $a \in Z_p^*$, $a^{p-1} \equiv 1 \mod p$.

1. Proof?
2. It gives an alternate way to find a^{-1}, how?

Properties of Prime

Fermat's little Theorem

Fix a prime p, for any $a \in Z_p^*$, $a^{p-1} \equiv 1 \mod p$.

1. Proof?
2. It gives an alternate way to find a^{-1}, how?
3. Can we use it to test primes? Say randomly pick a and compute a^{p-1}
4. Does it reject composites?

Properties of Prime

Fermat's little Theorem

Fix a prime p, for any $a \in Z_p^*$, $a^{p-1} \equiv 1 \mod p$.

1. Proof?
2. It gives an alternate way to find a^{-1}, how?
3. Can we use it to test primes? Say randomly pick a and compute a^{p-1}
4. Does it reject composites?
5. ☹ 561 is a counter-example: for any $gcd(a, 561) = 1$, $a^{560} \equiv 1 (\mod p)$

Properties of Prime

Fermat's little Theorem

Fix a prime p, for any $a \in Z_p^*$, $a^{p-1} \equiv 1 \mod p$.

1. Proof?
2. It gives an alternate way to find a^{-1}, how?
3. Can we use it to test primes? Say randomly pick a and compute a^{p-1}
4. Does it reject composites?
5. ☹ 561 is a counter-example: for any $gcd(a, 561) = 1$, $a^{560} \equiv 1 (\mod p)$

Properties of Prime

Fermat's little Theorem

Fix a prime p, for any $a \in Z_p^*$, $a^{p-1} \equiv 1 \mod p$.

1. Proof?
2. It gives an alternate way to find a^{-1}, how?
3. Can we use it to test primes? Say randomly pick a and compute a^{p-1}
4. Does it reject composites?
5. ☹ 561 is a counter-example: for any $gcd(a, 561) = 1$, $a^{560} \equiv 1(\mod p)$
6. Wait! How many a that has $gcd(a, 561) = 1$?

Properties of Prime

Fermat's little Theorem

Fix a prime p, for any $a \in Z_p^*$, $a^{p-1} \equiv 1 \mod p$.

1. Proof?
2. It gives an alternate way to find a^{-1}, how?
3. Can we use it to test primes? Say randomly pick a and compute a^{p-1}
4. Does it reject composites?
5. ☹ 561 is a counter-example: for any $gcd(a, 561) = 1$, $a^{560} \equiv 1(\mod p)$
6. Wait! How many a that has $gcd(a, 561) = 1$?
7. Let us introduce Euler function $\phi(N)$

Property: Prime Factorization

Unique Factorization

For any integer N, there is an unique factorization of primes $N = p_1^{e_1} \cdot p_2^{e_2} \cdots p_r^{e_R}$.

1. Definition of Euler Function: $\phi(N) = (1 - 1/p_1) \cdots (1 - 1/p_r) \cdot N$ if N's distinct prime factors are $p_1, \ldots, p_r$

Property: Prime Factorization

Unique Factorization

For any integer N, there is an unique factorization of primes $N = p_1^{e_1} \cdot p_2^{e_2} \cdots p_r^{e_R}$.

1. Definition of Euler Function: $\phi(N) = (1 - 1/p_1) \cdots (1 - 1/p_r) \cdot N$ if N's distinct prime factors are $p_1, \ldots, p_r$
2. Fact: Number of $a \in Z_N$ with $(a, N) = 1$ is $\phi(N)$

Property: Prime Factorization

Unique Factorization

For any integer N, there is an unique factorization of primes $N = p_1^{e_1} \cdot p_2^{e_2} \cdots p_r^{e_R}$.

1. Definition of Euler Function: $\phi(N) = (1 - 1/p_1) \cdots (1 - 1/p_r) \cdot N$ if N's distinct prime factors are $p_1, \ldots, p_r$
2. Fact: Number of $a \in Z_N$ with $(a, N) = 1$ is $\phi(N)$
3. Extension of Fermat's little THM: $a^{\phi(N)} \equiv 1 \mod N$ for any $(a, N) = 1$.

Composite Numbers

To distinguish prime numbers from composites, what properties does a composite have?

Chinese Remainder THM

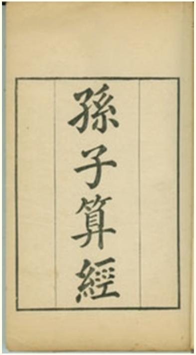
Given distinct primes $p_1, \ldots, p_r$, consider $(a_1, \ldots, a_r) \in Z_{p_1} \times \cdots \times Z_{p_r}$.
There is a 1-1 map between $a \in Z_{p_1 \cdot p_2 \cdots p_r}$ and $(a_1, \ldots, a_n)$

Composite Numbers

To distinguish prime numbers from composites, what properties does a composite have?

Chinese Remainder THM

Given distinct primes $p_1, \ldots, p_r$, consider $(a_1, \ldots, a_r) \in Z_{p_1} \times \cdots \times Z_{p_r}$. There is a 1-1 map between $a \in Z_{p_1 \cdot p_2 \cdots p_r}$ and $(a_1, \ldots, a_n)$



1. Most fundamental THM in group/ring theory and number theory
2. Both directions are useful in algorithm design and analysis

Outline

Basic Idea

Problem Description

Given q, test whether it is a prime or not.

Basic Idea

Problem Description

Given q, test whether it is a prime or not.

1. The issue of Fermat's little THM is that for some composites like $q = 561$, most $a \in Z_q$ has $a^{q-1} \equiv 1$

Basic Idea

Problem Description

Given q, test whether it is a prime or not.

1. The issue of Fermat's little THM is that for some composites like $q = 561$, most $a \in Z_q$ has $a^{q-1} \equiv 1$
2. Let us try Chinese Remainder THM.

Basic Idea

Problem Description

Given q, test whether it is a prime or not.

1. The issue of Fermat's little THM is that for some composites like $q = 561$, most $a \in Z_q$ has $a^{q-1} \equiv 1$
2. Let us try Chinese Remainder THM.

Key Observation

1. If q is a prime, $x^2 \equiv 1$ has at most 2 roots. What are they?
2. If q is not a prime, $x^2 \equiv 1$ has at least 4 roots. Why?

New Idea

Taking Square Root

If we can generate a random root of $x^2 \equiv 1 \mod q$, it tells whether q is a prime or not — prime q has at most 2 roots and composite q has $\geqslant 4$ roots

New Idea

Taking Square Root

If we can generate a random root of $x^2 \equiv 1 \mod q$, it tells whether q is a prime or not — prime q has at most 2 roots and composite q has $\geqslant 4$ roots

1. Unfortunately, finding square roots is computational hard — equivalent to factoring and break RSA system

New Idea

Taking Square Root

If we can generate a random root of $x^2 \equiv 1 \mod q$, it tells whether q is a prime or not — prime q has at most 2 roots and composite q has $\geqslant 4$ roots

1. Unfortunately, finding square roots is computational hard — equivalent to factoring and break RSA system
2. Wait ... we can use Fermat's little THM to help with it

New Idea

Taking Square Root

If we can generate a random root of $x^2 \equiv 1 \mod q$, it tells whether q is a prime or not — prime q has at most 2 roots and composite q has $\geqslant 4$ roots

1. Unfortunately, finding square roots is computational hard — equivalent to factoring and break RSA system
2. Wait ... we can use Fermat's little THM to help with it
3. Since $a^{q-1} \equiv 1$ for any a, $a^{\frac{q-1}{2}}$ might be a random root for a random a

New Idea

Taking Square Root

If we can generate a random root of $x^2 \equiv 1 \mod q$, it tells whether q is a prime or not — prime q has at most 2 roots and composite q has $\geqslant 4$ roots

1. Unfortunately, finding square roots is computational hard — equivalent to factoring and break RSA system
2. Wait ... we can use Fermat's little THM to help with it
3. Since $a^{q-1} \equiv 1$ for any a, $a^{\frac{q-1}{2}}$ might be a random root for a random a
    - If $a^{\frac{q-1}{2}} \neq \pm 1$, claim q is composite
    - If $a^{\frac{q-1}{2}} = -1$, give up this a

New Idea

Taking Square Root

If we can generate a random root of $x^2 \equiv 1 \mod q$, it tells whether q is a prime or not — prime q has at most 2 roots and composite q has $\geqslant 4$ roots

1. Unfortunately, finding square roots is computational hard — equivalent to factoring and break RSA system
2. Wait ... we can use Fermat's little THM to help with it
3. Since $a^{q-1} \equiv 1$ for any a, $a^{\frac{q-1}{2}}$ might be a random root for a random a
   - If $a^{\frac{q-1}{2}} \neq \pm 1$, claim q is composite
   - If $a^{\frac{q-1}{2}} = -1$, give up this a
   - Else if $a^{\frac{q-1}{2}} = 1$, keep trying $a^{\frac{q-1}{4}}, a^{\frac{q-1}{8}}, \ldots$

Formal Description

Pick many random $a \sim Z_q$ and call WITNESS(a) for each one — output "composite" if any call does so

---

Algorithm Miller-Rabin Tester

---

function WITNESS(a, q)
    Decompose $q - 1 = 2^t \cdot u$
    $x_0 \equiv a^u$
    for $i = 1, \ldots, t$ **do**
        $x_i \equiv x_{i-1}^2$ // x_{i-1} is a square root of x_i
        if $x_i \equiv 1$ and $x_{i-1} \neq \pm 1$ **then**
            Return Composite
    Return Composite if $x_t \neq 1$ o.w. return Prime

Formal Description

Pick many random $a \sim Z_q$ and call WITNESS(a) for each one — output "composite" if any call does so

---

Algorithm Miller-Rabin Tester

---

function WITNESS(a, q)
  Decompose $q - 1 = 2^t \cdot u$
  $x_0 \equiv a^u$
  for $i = 1, \ldots, t$ **do**
    $x_i \equiv x_{i-1}^2$ // x_{i-1} is a square root of x_i
    if $x_i \equiv 1$ and $x_{i-1} \neq \pm 1$ **then**
      Return Composite
  Return Composite if $x_t \neq 1$ o.w. return Prime

Running Time

$O(\log^3 q)$: $t = O(\log q)$ — but $x_i^2 \mod q$ takes $O(\log^2 q)$ time

Analysis

Correctness

If q is composite, at least half a make $\textsc{Witness}(a, q) = $ Composite.

The proof is similar to the proof $x^2 \equiv 1$ has at least 4 roots but uses group theory

Analysis

Correctness

If q is composite, at least half a make $\textsc{Witness}(a, q) =$ Composite.

The proof is similar to the proof $x^2 \equiv 1$ has at least 4 roots but uses group theory

1. Suppose $q = p_1 \cdot p_2$ for two distinct primes

Analysis

Correctness

If q is composite, at least half a make $\textsc{Witness}(a, q) = $ Composite.

The proof is similar to the proof $x^2 \equiv 1$ has at least 4 roots but uses group theory

1. Suppose $q = p_1 \cdot p_2$ for two distinct primes
2. Define (a, j) is acceptable if $a^{2^j \cdot u} \equiv -1$
3. Find the largest j with acceptable pairs (existence from $j = 0$ and $a = -1$) such that $x^{2^{j+1} u} \equiv 1$ for all x

Analysis

Correctness

If q is composite, at least half a make $\textsc{Witness}(a, q) = $ Composite.

The proof is similar to the proof $x^2 \equiv 1$ has at least 4 roots but uses group theory

1. Suppose $q = p_1 \cdot p_2$ for two distinct primes
2. Define (a, j) is acceptable if $a^{2^j \cdot u} \equiv -1$
3. Find the largest j with acceptable pairs (existence from $j = 0$ and $a = -1$) such that $x^{2^{j+1}u} \equiv 1$ for all x
4. But $B = \{x : x^{2^j u} \equiv \pm 1\} \subsetneq Z_q^*$ is a proper subgroup
5. By group theory, only need to show $\exists b \notin B$, which is implied by Chinese Remainder THM

Analysis

Correctness

If q is composite, at least half a make $\text{WITNESS}(a, q) = $ Composite.

The proof is similar to the proof $x^2 \equiv 1$ has at least 4 roots but uses group theory

1. Suppose $q = p_1 \cdot p_2$ for two distinct primes
2. Define (a, j) is acceptable if $a^{2^j \cdot u} \equiv -1$
3. Find the largest j with acceptable pairs (existence from $j = 0$ and $a = -1$) such that $x^{2^{j+1}u} \equiv 1$ for all x
4. But $B = \{x : x^{2^j u} \equiv \pm 1\} \subsetneq Z_q^*$ is a proper subgroup
5. By group theory, only need to show $\exists b \notin B$, which is implied by Chinese Remainder THM
6. When $i = j + 1$, half elements will be caught

Summary

1. Primality tester faster than $O(n^3)$?
2. How to generate primes efficiently?
3. Many other problem: factoring, ...

Questions?