Design and Analysis of Algorithms (CSC-314)

Department of B.Sc. CSIT
Godawari College

- Unit-7: Number Theoretic Algorithms

- **Concept of Number Theoretic Notation**
- Number theory was once viewed as a beautiful but largely useless subject in pure mathematics.
- Today number-theoretic algorithms are used widely, due in part to the invention of cryptographic schemes based on large prime numbers.
- The feasibility of these schemes rests on our ability to find large primes easily, while their security rests on our inability to factor the product of large primes.
- This chapter presents some of the number theory and associated algorithms that underlie such applications.

- Unit-7: Number Theoretic Algorithms

- **Concept of Number Theoretic Notation**
- Here we will discuss some useful properties of numbers when calculations are done modulo n, where n > 0.
- In the context of computer science, n is usually a power of 2 since representation is binary.

- Unit-7: Number Theoretic Algorithms

- **Concept of Number Theoretic Notation**

- Here it provides a brief review of notions from elementary number theory concerning
  - the set Z = {. . . , -2, -1, 0, 1, 2, . . .} of integers and
  - the set N = {0, 1, 2, . . .} of natural numbers.

• Unit-7: Number Theoretic Algorithms

- **Concept of Number Theoretic Notation**
- **Divisibility and divisors**
  - The notion of one integer being divisible by another is a central one in the theory of numbers.
  - The notation d | a (read "d divides a") means that a = kd for some integer k.
  - Every integer divides 0. If a > 0 and d | a, then |d| <= |a|.
  - If d | a, then we also say that a is a multiple of d.
  - If d | a and d> 0, we say that d is a divisor of a.
  - A divisor of an integer a is at least 1 but not greater than |a|. For example, the divisors of 24 are 1, 2, 3, 4, 6, 8, 12, and 24.
  - Every integer a is divisible by the trivial divisors 1 and a.
  - Nontrivial divisors of a are also called factors of a. For example, the factors of 20 are 2, 4, 5, and 10.

• Unit-7: Number Theoretic Algorithms

- **Concept of Number Theoretic Notation**
- **Prime and composite numbers**
  - An integer a > 1 whose only divisors are the trivial divisors 1 and a is said to be a prime number (or, more simply, a prime).
  - Primes have many special properties and play a critical role in number theory. The small primes, in order, are
    - 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59,... .
  - An integer a > 1 that is not prime is said to be a composite number (or, more simply, a composite).
  - For example, 39 is composite because 3 | 39.
  - The integer 1 is said to be a unit and is neither prime nor composite.
  - Similarly, the integer 0 and all negative integers are neither prime nor composite.

Unit-7: Number Theoretic Algorithms

- **Concept of Number Theoretic Notation**
- **Modular Division**
  - Given three positive numbers a, b and m. Compute a/b under modulo m. The task is basically to find a number c such that (b * c) % m = a % m.
  - **Examples**:
  - Input : a = 8, b = 4, m = 5
    - Output : 2
  - Input : a = 8, b = 3, m = 5
    - Output : 1
    - Note that (1*3)%5 is same as 8%5
  - Input : a = 11, b = 4, m = 5
    - Output : 4
    - Note that (4*4)%5 is same as 11%5

Unit-7: Number Theoretic Algorithms

- **Concept of Number Theoretic Notation**
- **Congruence modulo n**
  - Suppose $a,b,n \in Z$. We say a is congruent to b modulo n iff $a-b$ is divisible by n. The notation for this is: $a \equiv b \pmod{n}$.
  - **Examples**:
  - We have $22 \equiv 0 \pmod{2}$, because $22-0=22=11\times 2$ is a multiple of 2. (More generally, for $a \in Z$, one can show that $a \equiv 0 \pmod{2}$ iff a is even.)
  - We have $15 \equiv 1 \pmod{2}$, because $15-1=14=7\times 2$ is a multiple of 2. (More generally, for $a \in Z$, one can show that $a \equiv 1 \pmod{2}$ iff a is odd.)
  - We have $28 \equiv 13 \pmod{5}$, because $28-13=15=3\times 5$ is a multiple of 5.

Unit-7: Number Theoretic Algorithms

- **Concept of Number Theoretic Notation**
- **Greatest common divisor (GCD)**
- The greatest common divisor (GCD) refers to the greatest positive integer that is a common divisor for a given set of positive integers.
- It is also termed as the highest common factor (HCF) or the greatest common factor (GCF)
- For a set of positive integers (a, b), the greatest common divisor is defined as the greatest positive number which is a common factor of both the positive integers (a, b).
- GCD of any two numbers is never negative or 0 as the least positive integer common to any two numbers is always 1.
- There are some ways to determine the greatest common divisor of two numbers:
  - By finding the common divisors
  - By Euclid's algorithm

Unit-7: Number Theoretic Algorithms

- **Concept of Number Theoretic Notation**
- **How to Find the Greatest Common Divisor?**
- **For a set of two positive integers (a, b) we use the below-given steps to find the greatest common divisor:**
  - Step 1: Write the divisors of positive integer "a".
  - Step 2: Write the divisors of positive integer "b".
  - Step 3: Enlist the common divisors of "a" and "b".
  - Step 4: Now find the divisor which is the highest of both "a" and "b".
- **Example: Find the greatest common divisor of 13 and 48.**
  - Solution: We will use the below steps to determine the greatest common divisor of (13, 48).
  - Divisors of 13 are 1, and 13.
  - Divisors of 48 are 1, 2, 3, 4, 6, 8, 12, 16, 24 and 48.
  - The common divisor of 13 and 48 is 1.
  - The greatest common divisor of 13 and 48 is 1.
  - **Thus, GCD(13, 48) = 1.**

Unit-7: Number Theoretic Algorithms

- **Concept of Number Theoretic Notation**
- **Finding Greatest Common Divisor by LCM Method**
- As per the LCM Method for the greatest common divisor, the GCD of two positive integers (a, b) can be calculated by using the following formula:
  - **GCD (a, b) = (a × b)/ LCM(a,b)**
- **The steps to calculate the GCD of (a, b) using the LCM method is:**
  - Step 1: Find the product of a and b.
  - Step 2: Find the least common multiple (LCM) of a and b.
  - Step 3: Divide the values obtained in Step 1 and Step 2.
  - Step 4: The obtained value after division is the greatest common divisor of (a, b).
- **Example: Find the greatest common divisor of 15 and 70 using the LCM method.**
  - Solution: The greatest common divisor of 15 and 70 can be calculated as:
  - The product of 15 and 70 is given as, 15 × 70
  - The LCM of (15, 70) is 210.
  - GCD (15, 20) = (15 × 70)/ 210 = 5.
  - ∴ **The greatest common divisor of (15, 70) is 5.**

- **Concept of Number Theoretic Notation**
- **Euclid's Algorithm for Greatest Common Divisor**
  - As per Euclid's algorithm for the greatest common divisor, the GCD of two positive integers (a, b) can be calculated as:
    - If a = 0, then GCD (a, b) = b as GCD (0, b) = b.
    - If b = 0, then GCD (a, b) = a as GCD (a, 0) = a.
    - If both a≠0 and b≠0, we write 'a' in quotient remainder form (a = b×q + r) where **q** is the **quotient** and **r** is the **remainder**, and **a>b.**
  - Find the GCD (b, r) as GCD (b, r) = GCD (a, b)
  - We repeat this process until we get the remainder as 0.

Unit-7: Number Theoretic Algorithms

- **Concept of Number Theoretic Notation**
- **Example: Find the GCD of 12 and 10 using Euclid's Algorithm.**
  - Solution: The GCD of 12 and 10 can be found using the below steps:
  - $a = 12$ and $b = 10$, $a \neq 0$ and $b \neq 0$
  - In quotient remainder form we can write $12 = 10 \times 1 + 2$
  - Thus, GCD (10, 2) is to be found, as GCD(12, 10) = GCD (10, 2)
  - Now, $a = 10$ and $b = 2$,$a \neq 0$ and $b \neq 0$
  - In quotient remainder form we can write $10 = 2 \times 5 + 0$
  - Thus, GCD (2,0) is to be found, as GCD(10, 2) = GCD (2, 0)

- **Concept of Number Theoretic Notation**
- **Example: Find the GCD of 12 and 10 using Euclid's Algorithm.**
  - Now, a = 2 and b = 0, a≠0 and b=0
  - Thus, GCD (2,0) = 2
  - GCD (12, 10) = GCD (10, 2) = GCD (2, 0) = 2
  - Thus, GCD of 12 and 10 is 2.

  - Euclid's algorithm is very useful to find GCD of larger numbers, as in this we can easily break down numbers into smaller numbers to find the greatest common divisor.

- Unit-7: Number Theoretic Algorithms

- **Concept of Number Theoretic Notation**
- **Bézout’s theorem about GCDs**
  - **Bézout’s theorem**
  - If a and b are positive integers, then there exist integers u and v such that GCD(a, b) = ua + vb.
  - ***We can extend Euclidean algorithm to find u and v in addition to computing GCD(a, b).***

- Unit-7: Number Theoretic Algorithms

- **Concept of Number Theoretic Notation**

- Eg: $\gcd(1547, 560) = 7$

$$1547 = 2 \cdot 560 + 427$$
$$560 = 1 \cdot 427 + 133$$
$$427 = 3 \cdot 133 + 28$$
$$133 = 4 \cdot 28 + 21$$
$$28 = 1 \cdot 21 + 7$$
$$21 = 3 \cdot 7 + 0$$

$$au + bv = d = \gcd(a, b)$$

$$\begin{aligned}
7 &= 28 - 1 \cdot 21 \\
&= 28 - 1 \cdot (133 - 4 \cdot 28) \\
&= 5 \cdot 28 - 1 \cdot 133 \\
&= 5 \cdot (427 - 3 \cdot 133) - 1 \cdot 133 \\
&= 5 \cdot 427 - 16 \cdot 133 \\
&= 5 \cdot 427 - 16(560 - 1 \cdot 427) \\
&= 21 \cdot 427 - 16 \cdot 560 \\
&= 21(1547 - 2 \cdot 560) - 16 \cdot 560 \\
&= 21 \cdot 1547 - 58 \cdot 560 \\
&= u \cdot 1547 + v\, 560
\end{aligned}$$

$u = 21$
$v = -58$

- **Concept of Number Theoretic Notation**
- **Example :** Determine the greatest common divisor of 456 and 123 using Extended *Euclidean algorithm*.

Unit-7: Number Theoretic Algorithms

- **Concept of Number Theoretic Notation**
- **Solving Linear Equations Modulo n**
  - Consider $ax \equiv b \pmod{n}$
  - How can we find a solution to this equation without trying every possible value of x?
  - If $ax \equiv b \pmod{n}$, then $n \mid (b - ax)$ for some integer k, so $b - ax = nk$.
  - We are looking for values of k and x that satisfy the equation $b = nk + ax$.
  - Through previous investigation with the Euclidean Algorithm, we know that equations of the form $b = nk + ax$ have a solution if and only if $\gcd(a, n) \mid b$.
- **Theorem** . The equation $ax \equiv b \pmod{n}$ has a solution if and only if $\gcd(a, n) \mid b$. The solution to the equation is unique if and only if $\gcd(a, n) = 1$

Unit-7: Number Theoretic Algorithms

- **Concept of Number Theoretic Notation**
- **Solving Linear Equations Modulo n**
  - **Example 1: Solve 3x ≡ 5 (mod 6)**
    - Note that gcd (3, 6) = 3 and 3 - 5. Thus this equation has no solution.
  - **Example 2: Solve 3x ≡ 12 (mod 6)**
    - Note that gcd (3, 6) = 3 and 3 | 12. Thus this equation has solutions, but they are not unique since gcd (3, 6) 6 = 1.
    - x ≡ 2 (mod 6) since 3(2) ≡ 6 ≡ 12 (mod 6)
    - x ≡ 4 (mod 6) since 3(4) ≡ 12 (mod 6)
    - x ≡ 6 (mod 6) since 3(6) ≡ 18 ≡ 12 (mod 6)
  - **Example 3: Solve 5x ≡ 2 (mod 6)**
    - Solve this ?

Unit-7: Number Theoretic Algorithms

- **Concept of Number Theoretic Notation**
- **Solving Linear Equations Modulo n**
  - **Example 3: Solve 5x ≡ 2 (mod 6)**
  - Note that gcd (5, 6) = 1. Thus this equation has a solution and it is unique.
  - **X** 5x (mod 6)
  - 0 0 (mod 6)
  - 1 5 (mod 6)
  - 2 10 ≡ 4 (mod 6)
  - 3 15 ≡ 3 (mod 6)
  - **4 20 ≡ 2 (mod 6)**
  - 5 25 ≡ 1 (mod 6)
  - **Thus x ≡ 4 (mod 6) is the one unique solution.**

Unit-7: Number Theoretic Algorithms

- **Chinese Remainder Theorem:**

Statement:

- If m1,m2,… mk are relatively prime positive integers and if a1,a2,..ak are any integer then the simultaneous congruence's

x ≡ a1 (mod m1),

x ≡ a2 (mod m2),

…………………..

x ≡ ak (mod mk), have a solution and the solution is unique over modulo m.

Unit-7: Number Theoretic Algorithms

- **Chinese Remainder Theorem:**

Then we need to calculate x with following

- M = m1.m2. …… .mk
- Mi= M/mi
- X= (M1m1+M2m2+…….Mkmk)(mod M)
- MiXi =1 (mod mi)

Miller-Rabin Randomized Primility Test:

- Primality testing is an important algorithmic problem. In addition to being a fundamental mathematical question, the problem of how to determine whether a given number is prime has tremendous practical importance.
- Given an n-bit number N as input, we have to ascertain whether N is a prime number or not
- In practice, we use a randomized algorithm, namely, the Miller–Rabin Test, that successfully distinguishes primes from composites with very high probability.

Unit-7: Number Theoretic Algorithms

Miller-Rabin Randomized Primility Test: Algorithm

- It returns false if n is composite and returns true if n is probably prime. k is an input parameter that determines accuracy level. Higher value of k indicates more accuracy.
- bool isPrime(int n, int k)

  1) Handle base cases for n < 3

  2) If n is even, return false.

  3) Find an odd number d such that n-1 can be written as d*2r.

  Note that since n is odd, (n-1) must be even and r must be greater than 0.

  4) Do following k times

      if (millerTest(n, d) == false)

        return false

  5) Return true

Unit-7: Number Theoretic Algorithms

Miller-Rabin Randomized Primility Test: Algorithm

- This function is called for all k trials. It returns false if n is composite and returns true if n is probably prime. d is an odd number such that
  d*2r = n-1 for some r>=1

- bool millerTest(int n, int d)

  1) Pick a random number 'a' in range [2, n-2]

  2) Compute: x = pow(a, d) % n

  3) If x == 1 or x == n-1, return true.

  // Below loop mainly runs 'r-1' times.

  4) Do following while d doesn't become n-1.

  a) x = (x*x) % n.

  b) If (x == 1) return false.

  c) If (x == n-1) return true.

Miller-Rabin Randomized Primility Test:example

- Given Input: n = 13, k = 2.

1) Compute d and r such that d*2r = n-1,

    d = 3, r = 2.

2) Call millerTest k times.

Miller-Rabin Randomized Primility Test:example

- 1st Iteration:

  1) Pick a random number 'a' in range [2, n-2]

     Suppose a = 4

  2) Compute: x = pow(a, d) % n

     $x = 4^3 \% 13 = 12$

  3) Since x = (n-1), return true.

Miller-Rabin Randomized Primility Test:example

- 2nd Iteration:

  1) Pick a random number 'a' in range [2, n-2]

     Suppose a = 5

  2) Compute: x = pow(a, d) % n

     $x = 5^3 \% 13 = 8$

  3) x neither 1 nor 12.

  4) Do following (r-1) = 1 times

     a) x = (x * x) % 13 = (8 * 8) % 13 = 12

     b) Since x = (n-1), return true.

- Since both iterations return true, we return true.

Miller-Rabin Randomized Primility Test:example

- Use Miller-Rabin Randomized Primility Test to determine n=27 is prime or not?
- Use Miller-Rabin Randomized Primility Test to determine n=1729, using a = 671.

Unit-7: Number Theoretic Algorithms

- Thank You!