UNIT-II

Contents

- Introduction to Number Theory
- Fermat’s and Euler’s Theorem
- The Chinese Remainder Theorem
- Euclidean Algorithm, and
- Modular Arithmetic

Prime Numbers

- prime numbers only have divisors of 1 and self
  - they cannot be written as a product of other numbers
  - note: 1 is prime
- eg. 2,3,5,7 are prime, 4,6,8,9,10 are not
- prime numbers are central to number theory
- list of prime number less than 200 is:

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59
61 67 71 73 79 83 89 97 101 103 107 109 113 127
131 137 139 149 151 157 163 167 173 179 181 191
193 197 199
```

Prime Factorisation

- to **factor** a number n is to write it as a product of other numbers: $n = a \times b \times c$
- note that factoring a number is relatively hard compared to multiplying the factors together to generate the number
- the **prime factorisation** of a number n is when its written as a product of primes
  - eg. $91 = 7 \times 13$; $3600 = 2^4 \times 3^2 \times 5^2$
  - It is unique $a = \prod_{p \in P} p^{a_p}$

Relatively Prime Numbers & GCD

- two numbers `a, b` are **relatively prime** if have **no common divisors** apart from 1
  - eg. 8 & 15 are relatively prime since factors of 8 are 1,2,4,8 and of 15 are 1,3,5,15 and 1 is the only common factor
- conversely can determine the greatest common divisor by comparing their prime factorizations and using least powers
  - eg. $300=2^1\times3^1\times5^2$ $18=2^1\times3^2$ hence $GCD(18,300)=2^1\times3^1\times5^0=6$

Fermat's Little Theorem

- $a^{p-1} \bmod p = 1$

  where p is prime and a is a positive integer not divisible by p
- also known as Fermat’s Little Theorem
- useful in public key and primality testing

Euler Totient Function ø(n)

- when doing arithmetic modulo n
- **complete set of residues** is: `0..n-1`
- **reduced set of residues** includes those numbers which are relatively prime to n
  - eg for n=10,
  - complete set of residues is {0,1,2,3,4,5,6,7,8,9}
  - reduced set of residues is {1,3,7,9}
- **Euler Totient Function ø(n):**
  - number of elements in reduced set of residues of n
  - **ø(10) = 4**

Euler Totient Function $\varnothing(n)$

- to compute ø(n) need to count number of elements to be excluded
- in general need prime factorization, but
  - for p (p prime) $\varnothing(p) = p-1$
  - for p.q (p,q prime) $\varnothing(p.q) = (p-1)(q-1)$
- eg.
  - $\varnothing(37) = 36$
  - $\varnothing(21) = (3-1)\times(7-1) = 2\times6 = 12$

Euler's Theorem

- a generalisation of Fermat's Theorem
- $a^{\varnothing(n)} \bmod n = 1$
  - where $\gcd(a,n)=1$
- eg.
  - $a=3; n=10; \varnothing(10)=4;$
  - hence $3^4 = 81 = 1 \bmod 10$
  - $a=2; n=11; \varnothing(11)=10;$
  - hence $2^{10} = 1024 = 1 \bmod 11$

Chinese Remainder Theorem

- Used to speed up modulo computations
- Used to modulo a product of numbers
  - eg. mod $M = m_1 m_2 .. m_k$, where $\gcd(m_i, m_j)=1$
- Chinese Remainder theorem lets us work in each moduli m_i separately
- since computational cost is proportional to size, this is faster than working in the full modulus M

Chinese Remainder Theorem

- to compute (A mod M) can firstly compute all (a_i mod m_i) separately and then combine results to get answer using:

$$A \equiv \left(\sum_{i=1}^{k} a_i c_i \right) \bmod M$$

$$c_i = M_i \times \left(M_i^{-1} \bmod m_i \right) \quad \text{for } 1 \le i \le k$$

Divisors

- say a non-zero number `b` **divides** `a` if for some `m` have `a=mb` (`a,b,m` all integers)
- that is `b` divides into `a` with no remainder
- denote this `b|a`
- and say that `b` is a **divisor** of `a`
- eg. all of 1,2,3,4,6,8,12,24 divide 24
- eg. 13 | 182; –5 | 30; 17 | 289; –3 | 33; 17 | 0

Properties of Divisibility

- If $a|1$, *then* $a = \pm 1$.
- *If* $a|b$ *and* $b|a$, *then* $a = \pm b$.
- *Any* b /= 0 *divides* 0.
- *If* $a \mid b$ *and* $b \mid c$, *then* $a \mid c$
  - *e.g.* 11 | 66 and 66 | 198 x 11 | 198
- If $b|g$ *and* $b|h$, *then* $b|(mg + nh)$

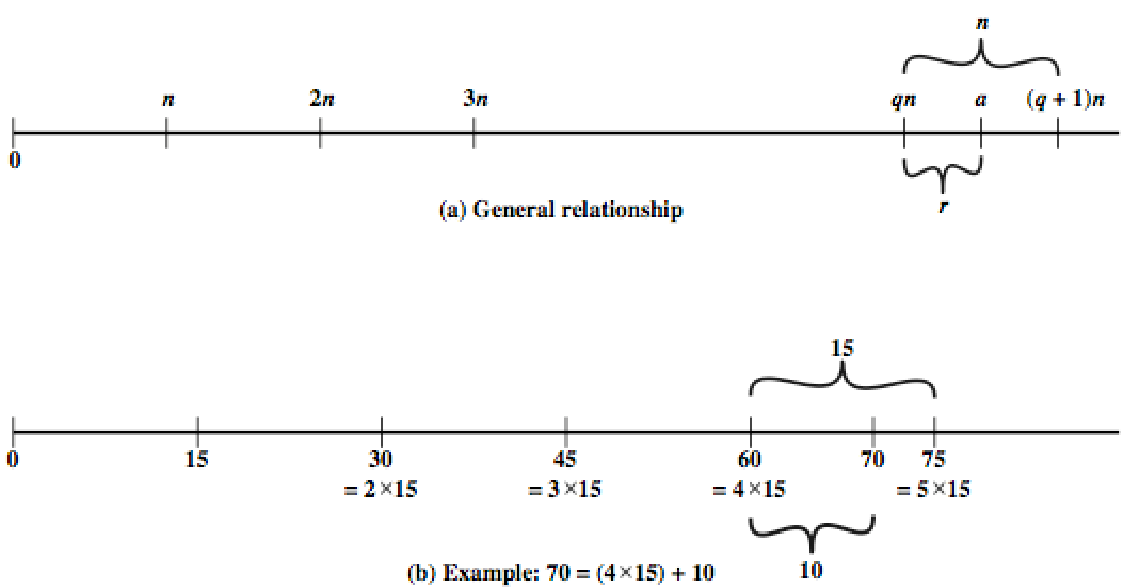
  for arbitrary integers m *and* n

  e.g. $b = 7$; $g = 14$; $h = 63$; $m = 3$; $n = 2$

  hence $7|14$ *and* $7|63$

Division Algorithm

- if divide a by n get integer quotient *q* and integer remainder *r* such that:
  - $a = qn + r$ where $0 <= r < n$; $q = floor(a/n)$
- remainder *r* often referred to as a **residue**



(a) General relationship

(b) Example: 70 = (4×15) + 10

Greatest Common Divisor (GCD)

- a common problem in number theory
- GCD (a,b) of a and b is the largest integer that divides evenly into both a and b
  - eg GCD(60,24) = 12
- define gcd(0, 0) = 0
- often want **no common factors** (except 1) define such numbers as **relatively prime**
  - eg GCD(8,15) = 1
  - hence 8 & 15 are relatively prime

Example GCD(1970,1066)

```
1970 = 1 x 1066 + 904     gcd(1066, 904)
1066 = 1 x 904 + 162      gcd(904, 162)
904 = 5 x 162 + 94  gcd(162, 94)
162 = 1 x 94 + 68         gcd(94, 68)
94 = 1 x 68 + 26          gcd(68, 26)
68 = 2 x 26 + 16          gcd(26, 16)
26 = 1 x 16 + 10          gcd(16, 10)
16 = 1 x 10 + 6           gcd(10, 6)
10 = 1 x 6 + 4      gcd(6, 4)
6 = 1 x 4 + 2       gcd(4, 2)
4 = 2 x 2 + 0       gcd(2, 0)
```

GCD(1160718174, 316258250)

Dividend	Divisor	Quotient	Remainder
a = 1160718174	b = 316258250	q1 = 3	r1 = 211943424
b = 316258250	r1 = 211943424	q2 = 1	r2 = 104314826
r1 = 211943424	r2 = 104314826	q3 = 2	r3 = 3313772
r2 = 104314826	r3 = 3313772	q4 = 31	r4 = 1587894
r3 = 3313772	r4 = 1587894	q5 = 2	r5 = 137984
r4 = 1587894	r5 = 137984	q6 = 11	r6 = 70070
r5 = 137984	r6 = 70070	q7 = 1	r7 = 67914
r6 = 70070	r7 = 67914	q8 = 1	r8 = 2516
r7 = 67914	r8 = 2516	q9 = 31	r9 = 1078
r8 = 2516	r9 = 1078	q10 = 2	r10 = 0

Modular Arithmetic

- define **modulo operator** "`a mod n`" to be remainder when a is divided by n
  - where integer *n* is called the **modulus**
- *b* is called a **residue** of *a* mod *n*
  - since with integers can always write: `a = qn + b`
  - usually chose smallest positive remainder as residue
    - ie. `0 <= b <= n-1`
  - process is known as **modulo reduction**
    - `eg. -12 mod 7 = -5 mod 7 = 2 mod 7 = 9 mod 7`
- *a* & *b* are **congruent** if: `a mod n = b mod n`
  - when divided by *n,* a & b have same remainder
  - eg. 100 = 34 mod 11

Modular Arithmetic Operations

- can perform arithmetic with residues
- uses a finite number of values, and loops back from either end

  $Z_n = \{0, 1, . . . , (n – 1)\}$
- modular arithmetic is when do addition & multiplication and modulo reduce answer
- can do reduction at any point, ie
  - `a+b mod n = [a mod n + b mod n] mod n`

Modular Arithmetic Operations

1. $[(a \bmod n) + (b \bmod n)] \bmod n = (a + b) \bmod n$
2. $[(a \bmod n) - (b \bmod n)] \bmod n = (a - b) \bmod n$
3. $[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$

e.g.

[(11 mod 8) + (15 mod 8)] mod 8 = 10 mod 8 = 2 (11 + 15) mod 8 = 26 mod 8 = 2

[(11 mod 8) – (15 mod 8)] mod 8 = –4 mod 8 = 4 (11 – 15) mod 8 = –4 mod 8 = 4

[(11 mod 8) x (15 mod 8)] mod 8 = 21 mod 8 = 5 (11 x 15) mod 8 = 165 mod 8 = 5

Modulo 8 Addition Example

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	0
2	2	3	4	5	6	7	0	1
3	3	4	5	6	7	0	1	2
4	4	5	6	7	0	1	2	3
5	5	6	7	0	1	2	3	4
6	6	7	0	1	2	3	4	5
7	7	0	1	2	3	4	5	6

Modulo 8 Multiplication

+	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	0	2	4	6
3	0	3	6	1	4	7	2	5
4	0	4	0	4	0	4	0	4
5	0	5	2	7	4	1	6	3
6	0	6	4	2	0	6	4	2
7	0	7	6	5	4	3	2	1

Modular Arithmetic Properties

Property	Expression
Commutative laws	$(w + x) \bmod n = (x + w) \bmod n$ $(w \times x) \bmod n = (x \times w) \bmod n$
Associative laws	$[(w + x) + y] \bmod n = [w + (x + y)] \bmod n$ $[(w \times x) \times y] \bmod n = [w \times (x \times y)] \bmod n$
Distributive law	$[w \times (x + y)] \bmod n = [(w \times x) + (w \times y)] \bmod n$
Identities	$(0 + w) \bmod n = w \bmod n$ $(1 \times w) \bmod n = w \bmod n$
Additive inverse $(-w)$	For each $w \in Z_n$, there exists a z such that $w + z = 0 \bmod n$

Euclidean Algorithm

- an efficient way to find the GCD(a,b)
- uses theorem that:
  - `GCD(a,b) = GCD(b, a mod b)`
- Euclidean Algorithm to compute GCD(a,b) is:

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

Extended Euclidean Algorithm

- calculates not only GCD but x & y:

  $ax + by = d = \gcd(a, b)$

- useful for later crypto computations
- follow sequence of divisions for GCD but assume at each step i, can find x &y:

  $r = ax + by$

- at end find GCD value and also x & y
- if GCD(a,b)=1 these values are inverses

Finding Inverses

EXTENDED EUCLID(m, b)

1. (A1, A2, A3)=(1, 0, m);
(B1, B2, B3)=(0, 1, b)

2. **if** B3 = 0
return A3 = gcd(m, b); no inverse

3. **if** B3 = 1
return B3 = gcd(m, b); B2 = b^{-1} mod m

4. Q = A3 div B3

5. (T1, T2, T3)=(A1 – Q B1, A2 – Q B2, A3 – Q B3)

6. (A1, A2, A3)=(B1, B2, B3)

7. (B1, B2, B3)=(T1, T2, T3)

8. **goto** 2

Inverse of 550 in GF(1759)

Q	A1	A2	A3	B1	B2	B3
—	1	0	1759	0	1	550
3	0	1	550	1	–3	109
5	1	–3	109	–5	16	5
21	–5	16	5	106	–339	4
1	106	–339	4	–111	355	1