Discrete Structures II

Mwangi H. (Ph.D.)

CS Yr 3.1
Department of Computing
J.K.U.A.T.

February 29, 2024

1. Divisibility and Modular Arithmetic
    - Divisibility
    - Division with Remainder
    - Congruence Relations
2. Greatest Common Divisors and Primes
    - Prime Numbers
    - Finding Primes
    - Computing Primes
    - Euclidean Algorithm
3. Solving Congruences
    - Linear Congruences
    - Systems of Linear Congruences
    - Exercises
4. Integer Representations
    - Positional Number System
    - Converting to radix-r
    - Binary, Octal, Hex Conversion
    - Binary Arithmetic

- This chapter discusses introduction to number theory, the study of integers and their properties
- Key considerations in number theory are the division and prime numbers which extends to modular arithmetic, and even cryptography.
- Number theory is typically a very “pure” mathematical topic, but it has many practical applications.
- Cryptography is very useful in practice, including internet security, banking, and more.
- Number theory also has many other applications in computer science.

- In this section we will explore the properties of integers $\ldots, -4, -3, -2, -1, 0, 1, 2, 3, 4, \ldots$
- In so-called ring theory the integers are an integral domain. The key property of integral domain is the cancellation property.
- If $a \neq 0$ and $ab = ac$, then $b = c$
- This property suggests a natural setting for the study of a divisibility.

Definition 1.1

If a and b are integers, with $a \neq 0$,we say a divides b if there exists an integer q such that $b = aq$. When a divides b we write $a|b$, otherwise $a \nmid b$

- From these definitions we get special names for a, b, q. When we have $b = aq$
  1. a is the divisor or factor of b
  2. b is the divided or multiple of a
  3. q is the quotient
- $3|9, 3 \nmid 17, 3|15$
- The property of divisibility leads to many combinations and later results. Important Divisibility Theorems.

Theorem 1.1

let a,b and c be integers with $a \neq 0$

- a. if $a|b$ and $a|c$ then $a|(b+c)$
- b. if $a|b$ then $a|bc$
- c. if $a|b$ and $b|c$ then $a|c$

Proof of the first statement

Proof.

Suppose $a|b$ and $a|c$. Then there exists integers $q1$ and $q2$ such that $b = aq1$ and $c = aq2$. Hence, $b + c = aq1 + aq2$. Therefore $b + c = a(q1 + q2)$. Since $q1 + q2$ is an integer, $a|(b + c)$ □

Proof the second and third statement of the theorem

- Division with remainder is also called *Euclidean Division*. It is both an algorithm and a theorem for computing quotients and remainders
- Recall that when a number divides another number perfectly, then we get a quotient and an equation of the form $b = aq$.
- However, it is often the case that division cannot be performed exactly. This is the role of the remainder.
- Example $3 \nmid 14$. Therefore $14 = 3.4 + 2$. We say that 3 divides 14, four times with a remainder of 2

Theorem 1.2: Euclidean Division

Let a,b be integers with $a \neq 0$. There exist unique integers q and r such that $b = aq + r$, and $0 \leq r < a$

Proof

Notice that we always re-write a division with remainder relation in terms of positive integers. Indeed if $a < 0$ then $b = aq + r$ can be re-written as $b = a'q' + r$ with $a' = -a$ and $q' = -q$. The case of $b < 0$ is similar. Therefore, we only have to consider the case where a, b, q, r are all non-negative integers. Now, we prove the existence of quotient and remainder. We will show $q = \lfloor \frac{b}{a} \rfloor$. By definition we have that $0 \leq r < a$ and $b = aq + r$ Therefore

$$
\begin{aligned}
aq &\leq aq + r < a(q+1) \\
aq &\leq b < a(q+r) \\
q &\leq \frac{b}{a} < q + 1 \\
q &= \lfloor \frac{b}{a} \rfloor
\end{aligned}
$$

Proof Cont'd

Since a, b, q, r are integers with $a \neq 0$ and $0 \leq r < b$, r can be uniquely be determined from a, b, and q Next we prove that the quotient and remainder are unique. Let $b = aq1 + r1 = aq2 + r2$ such that a, b, q, r are non-negative integers and $0 \leq r1 < a$ and $0 \leq r2 < a$. Proceed by contradiction and assume $r1 \neq r2$. W.L.O.G. assume $r2 > r1$. Then

$$aq1 - aq2 = r2 - r1$$
$$a(q1 - q2) = r2 - r1$$

From $a(q1 - q2) = r2 - r1$ we have that $a|r2 - r1$ However, since $0 \leq r1 < a$ and $0 \leq r2 < a$, and $r2 > r1$ it must be that $0 < (r2 - r1) < a$

Proof Cont'd

Yet, the multiples of a are $0, \pm a, \pm 2a, \pm 3a, \ldots$. Since $r2 - r1 < a$. it must be that $r2 - r1 = 0$ and hence $r2 = r1$. A contradiction. Since $r2 = r1$, it follows that $q2 = q1$ from $a \neq 0$ and

$$
\begin{aligned}
&aq1 + r1 = aq2 + r2 \\
&aq1 = aq2 \\
&a(q1 - q2) = 0 \implies q1 - q2 = 0
\end{aligned}
$$

□

- From Euclidean division, we get two sub-operations: div and mod. Div refers to the quotient and mod to the remainder
  1. b div $a = q$ and
  2. b mod $a = r$

- Congruence Relations or Congruences are special kinds of equivalences.
- Congruence is very similar to modulus

Definition 1.2

Two integers a and b are congruent modulo a positive integer m if m divides $a - b$

- Congruences are all about remainders. Modulo m, two integers a and b are congruent if $a \bmod m = b \bmod m$
- When two numbers are congruent modulo m, we write $a \equiv b \bmod m$ and we say "a is congruent to b modulo m". The relation $a \equiv b$ is a congruence and m is the modulus.

- Examples of Modulo 6: $17 \equiv 5 mod 6$
- One way to this of modulo is an operation which removes all multiples of the modulus from an expression. It is a simplification
- Consider 26 mod 5.
  $26 = 5.5 + 1$
- By Euclidean division, this implies that the remainder of 26 when divided by 5 is 1. Hence $26 \equiv 1 \text{ mod } 5$

Theorem 1.3

Let m be a positive integer. The integers a and b are congruent modulo m if and only if there exists and integer k such that $a = k.m + b$

Proof.

If $a \equiv b \bmod m$ then, by definition $m|a-b$. Hence, there exist an integer k such $a-b=km$. Rearranging, $a=b+km$. Conversely, if there exists and integer k such that $a=b+km$, then we of course have $km=a-b$ and thus $m|a-b$ and $a \equiv b \bmod m$ □

mod versus mod

Note that **mod** and mod are not exactly the same thing. Although highly related, one is a function $a \bmod b = r$ is a function taking a and b as arguments and returning the remainder r of a divided by b while the other is a binary relation. It defines an equivalence relation.

Congruence, Sums and Products

Congruence relations allow for interesting algebraic manipulation.

Theorem 1.4

Let a, b, c, d be integers, and let m be a positive integer. If $a \equiv b \mod m$ and $c \equiv d \mod m$ then $a + c \equiv b + d \mod m$

Proof.

By a previous theorem, we have that the congruences implies existence of integers k, l and the equations $b = a + km$ and $c = d + lm$ results. From these equations we have $b + d = (a + km) + (c + lm) = (a + c) + (k + l)m$. Therefore $b + d \equiv a + c \mod m$ □

Congruence, Sums and Products

- More generally, we can perform algebraic manipulations on several equations while "working modulo m".
- As a consequence of theorem above, if we multiply both sides of a congruence by the same integer, the congruence still holds.
- For any integer c we have $a \equiv b \bmod m \implies c.a \equiv c.b \bmod m$
- As a caution, division does not always maintain congruences. Notice $14 \equiv 8 \bmod 6$. However dividing by 2: $\frac{14}{2} \not\equiv \frac{8}{2} \bmod 6$

Congruence, Sums and Products -Arithmetic Modulo m

- Two important properties derived from the above theorems for modulo m are:
  1. $(a + b) \text{ mod } m = ((a \text{ mod } m) + (b \text{ mod } m)) \text{ mod } m$
  2. $(a.b) \text{ mod } m = ((a \text{ mod } m).(b \text{ mod } m)) \text{ mod } m$
- These equations have a very important consequence
- They mean we can perform either arithmetic first, and then take remainders or we can take remainders first and then perform arithmetic
- Let $\mathbb{Z} = 0, 1, 2, ...m - 1$ be the set of non-negative integers less that m. We can define a special kind of addition and multiplication using the above properties.
- These addition and multiplication operations act on elements of $\mathbb{Z}_m$ and always return another element of $\mathbb{Z}_m$

Congruence, Sums and Products - Arithmetic Modulo m

- (**Addition modulo** m)Let $+_m$ be addition on the set of numbers $\mathbb{Z}_m$ defined as $a +_m b = (a + b)\text{mod } m$
- (**Multiplication mudulo** m) Let $\times_m$ be multiplication on the set of numbers $\mathbb{Z}_m$ be defined as $a \times_m b = (a \times b)$ mod m
- $+_m$ and $\times_m$ rather than the normal addition and subtraction of the integers is called "working modulo m" or "doing arithmetic modulo m"
- Working modulo 11. Let $\mathbb{Z}_{11}$ be the set $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Then
- $7 +_{11} 9 = (7 + 9) \text{ mod } 11 = 16 \text{ mod } 11 = 5$
- $7 \times_{11} 9 = (7 \times 9) \text{ mod } 11 = 63 \text{ mod } 11 = 8$

Congruence, Sums and Products -Exercises

1. let $m = 17$ Compute $16 +_{17} 13$
2. let $m = 103$ Compute $45 \times_{103} 77$
3. let $m = 32$ Compute $4 \times_{32} 8$
4. let $m = 25$ Compute $-103 +_{25} 13$

Properties of arithmetic modulo m

- The operations of sum and product of modulo m are very similar to the normal addition and multiplication of the integers.
- These operations define a commutative ring or the ring theory

Property	Description
Closure	If $a, b \in \mathbb{Z}_m$, then $a +_m b \in \mathbb{Z}_m$ and $a \times_m b \in \mathbb{Z}_m$
Associativity	If $a, b, c \in \mathbb{Z}_m$, then $(a +_m b) +_m c = a +_m (b +_m c)$ and $(a \times_m b) \times_m c = a \times_m (b \times_m c)$
Commutativity	If $a, b \in \mathbb{Z}_m$, then $a +_m b = b +_m a$ and $a \times_m b = b \times_m a$
Identity	The elements 0 and 1 are identities of addition and multiplication. $a +_m 0 = a$ and $a \times_m 1 = a$
Additive inverses	If $a \neq 0 \in \mathbb{Z}_m$ then $m - a$ is the additive inverse of a such that $a +_m (m - a) = 0$
Distributivity	If $a, b, c \in \mathbb{Z}_m$ then $a \times_m (b +_m c) = (a \times_m b) + (a \times_m c)$

Congruences as equivalence relations

- Over the integers, congruence modulo m induces an equivalence relation on $\mathbb{Z}$
- As a binary relation on $\mathbb{Z}$, congruence mudulo m is Reflexive, Symmetric and Transitive
  1. **Reflexive**. For any $a \in \mathbb{Z}, a \equiv a \mod m$ since $m|a - a$
  2. **Symmetric**. For $a, b \in \mathbb{Z}, a \equiv b \mod m$ implies $b \equiv a \mod m$ since $m|a - b$ also implies $m|b - a$
  3. **Transitive**. If $a, b, c \in \mathbb{Z}$ with $a \equiv b \mod m$ and $b \equiv c \mod m$ then $a \equiv c \mod m$. Indeed, we have $m|(b - a)$ and $m|b - c$, thus $\exists k, l \in \mathbb{Z}$ such that: $b - a = mk$ and $c - b = ml \implies c - ml - a = mk$, rearranging $c - a = m(l + k)$. Hence $m|c - a$ and $a \equiv c \mod m$
- Since congruence is a an equivalence relation, it also implies the existence of equivalence classes. These are also called congruence classes.

Congruences as equivalence relations - Congruence classes

Definition 1.3: Congruence Class

The congruence class modulo m of an integer x is the set of all integers congruent to x modulo m

- Denote congruence classes as $\bar{x}$. This can also be denoted by typical equivalence class notation $\{x\}$. The set $\bar{x}$ is $\bar{x} = \{a \in \mathbb{X} | x \equiv a \bmod m\}$
- Notice that modulo itself does not define a congruence relation or congruence classes. A specific modulo need to be worked on.
- The set of all congruence classes, and the members of each congruent class, changes with the choice of modulus

Congruences as equivalence relations - Congruence classes

- $m = 5$ The set of all equivalence classes modulo m are $\{\bar{0}, \bar{1}, \bar{2}, \bar{3}, \bar{4}\}$
- This set is similar to $\mathbb{Z}_5 = \{0, 1, 2, 3, 4\}$. The set of congruence classes modulo 5 and $\mathbb{Z}_5$ are essentially the same. In other words they are isomorphic

Proposition 1.1: Proposition

An integer is congruent modulo m to its remainder on division by m. There are m congruence classes modulo m, each corresponding to the m possible remainders

Congruences as equivalence relations - Congruence classes

- Let $_m\mathbb{Z}$ represent all integers multiples of m. That is, the set $\{\cdots, -2m, -m, 0, m, 2m \cdots\}$. Then, the set of congruence classes modulo m are:

$$\begin{aligned}
\overline{0} &=_m \mathbb{X} \\
\overline{1} &=_m \mathbb{X} + 1 \\
\overline{2} &=_m \mathbb{X} + 2 \\
. & \\
. & \\
. & \\
\overline{m-1} &=_m \mathbb{X} + (m+1)
\end{aligned}$$

Congruence - Exercises

1. Proof the following theorem

   Theorem 1.5

   let a, b, c, d be integers and m be a positive integer. If $a \equiv b \mod m$ and $c \equiv d \mod m$ the $ac \equiv bd \mod m$

2. Prove that an integer is congruent modulo m to its remainders on division by m
3. Compute the following values
   - a. The quotient of 54 divided by 6
   - b. The remainder of 54 divided by 6
   - c. The Quotient of 1235 divided by 12
   - d. 144 mod 7
   - e. 123 mod 7
   - f. -17 mod 3
   - g. -101 mod 13

Congruence - Exercise

1. Consider an analog clock which shows the numbers 1 through 12. What time does it show :
   1. 48 hours after it shows 5:00?
   2. 17 hours after it shows 11:00?
   3. 103 hours after it shows 4:00?
2. Find an integer satisfying the following
   1. $x \equiv 43 \bmod 23$ where $-22 \leq x \leq 0$
   2. $x \equiv 17 \bmod 29$ where $-14 \leq x \leq 14$
   3. $x \equiv -11 \bmod 21$ where $90 \leq x \leq 110$

- GCD and prime numbers are a fundamental part of number theory.
- They have extensively been studied for years. Euclid was fundamental to the study of prime numbers and the number theory.

Definition 2.1: Prime

An integer $p > 1$ is prime if the only divisors of p are 1 and p. An integer which is not prime is called composite

- Examples
  - 7 is prime because only 1 and 7 divide 7. On the other hand, 3 divides 9 and so 9 is not prime
- Recall that another way of describing divisors is as factors. Therefore, an equivalent definition of a prime number p is one whose factors are only 1 and p
- The *fundamental theorem of arithmetic* strengthens the idea of primarity to a prime factorization

Theorem 2.1: Fundamental theorem of Arithmetic

Every integer greater that 1 is either prime or can be written as the product of two or more primes.
Formally, the theorem can be stated as follows. For Every integer $c > 1$ there exists a positive integer n, prime numbers $p_1, \cdots, p_n$ and exponents $e_1, \cdots, e_n$ such that

$$c = p_1^{e_1} p_2^{e_2} \cdots p_n^{e_n}$$

- This theorem is also called the *Unique factorization theorem*. It means that any number which is not prime is the product of some primes
  - a. $6 = 2.3$
  - b. $16 = 2.2.2.2.2 = 2^4$
  - c. $42 = 2.3.7$
  - d. $1234 = 2.617$
  - e. $1008 = 2.2.2.2.3.3.7 = 2^4.3^2.7$

- Since multiplication is commutative, prime factorization is only unique to the ordering of factors

$$420 = 2^2.3.5.7 = 7.3.5.2^2$$

- To get a unique prime factorization, we often add an additional constraint to the fundamental theorem of arithmetic. This constraint requires the primes to be listed in increasing order: $p_1 < p_2 \cdots < p_n$

- How can we determine if a numbers is prime?
- A Simple and *brute force* solution is to try and divide the number in question by every other integer less than it. If there are no divisors, then the number is prime. This methods is very inefficient though.
- The *Sieve of Eratosthenes* is a more efficient method. It is based on the following observation:

Proposition 2.1

If a positive integer n is composite, then it must have a prime divisor less than or equal to $\sqrt{n}$.

Proof.

If n is a positive composite then there exists two integers a, b greater than 1 such that $n = ab$. Certainly $a \leq \sqrt{n}$ or $b \leq \sqrt{n}$. If n is a perfect square then $a = b = \sqrt{n}$. Otherwise, one of a or b must be smaller than $\sqrt{n}$. □

- *Sieve of Eratosthenes* uses this proposition to remove all composite numbers from a list and retain only the prime ones
- Let $S = \{2, 3, \cdots, 100\}$. Since the maximum element of S is 100, we only need to consider prime divisors less than $\sqrt{100} = 10$.
  1. Find the smallest element of S. This is 2. This element is prime. Remove from S all multiples of 2 other than 2 itself.

- *Sieve of Eratosthenes* uses this proposition to remove all composite numbers from a list and retain only the prime ones
- Let $S = \{2, 3, \cdots, 100\}$. Since the maximum element of S is 100, we only need to consider prime divisors less than $\sqrt{100} = 10$.
  1. Find the smallest element of S. This is 2. This element is prime. Remove from S all multiples of 2 other than 2 itself.
  2. Find then next smallest element of the remaining numbers. This is 3. Remove from S all multiples of 3 other than 3 itself.

- *Sieve of Eratosthenes* uses this proposition to remove all composite numbers from a list and retain only the prime ones
- Let $S = \{2, 3, \cdots, 100\}$. Since the maximum element of S is 100, we only need to consider prime divisors less than $\sqrt{100} = 10$.
  1. Find the smallest element of S. This is 2. This element is prime. Remove from S all multiples of 2 other than 2 itself.
  2. Find then next smallest element of the remaining numbers. This is 3. Remove from S all multiples of 3 other than 3 itself.
  3. Find then next smallest element of the remaining numbers. This is 5. Remove from S all multiples of 5 other than 5 itself.

- *Sieve of Eratosthenes* uses this proposition to remove all composite numbers from a list and retain only the prime ones
- Let $S = \{2, 3, \cdots, 100\}$. Since the maximum element of S is 100, we only need to consider prime divisors less than $\sqrt{100} = 10$.
  1. Find the smallest element of S. This is 2. This element is prime. Remove from S all multiples of 2 other than 2 itself.
  2. Find then next smallest element of the remaining numbers. This is 3. Remove from S all multiples of 3 other than 3 itself.
  3. Find then next smallest element of the remaining numbers. This is 5. Remove from S all multiples of 5 other than 5 itself.
  4. Find then next smallest element of the remaining numbers. This is 7. Remove from S all multiples of 7 other than 7 itself.

- *Sieve of Eratosthenes* uses this proposition to remove all composite numbers from a list and retain only the prime ones
- Let $S = \{2, 3, \cdots, 100\}$. Since the maximum element of S is 100, we only need to consider prime divisors less than $\sqrt{100} = 10$.
  1. Find the smallest element of S. This is 2. This element is prime. Remove from S all multiples of 2 other than 2 itself.
  2. Find then next smallest element of the remaining numbers. This is 3. Remove from S all multiples of 3 other than 3 itself.
  3. Find then next smallest element of the remaining numbers. This is 5. Remove from S all multiples of 5 other than 5 itself.
  4. Find then next smallest element of the remaining numbers. This is 7. Remove from S all multiples of 7 other than 7 itself.
  5. Find next smallest element of S is 11. Since $11 > 10$, we can stop. Every remaining number is prime. Every remaining number is prime. The prime numbers less than 100 are 2,3,7,11,19,23,29,31,37,41,43,47,53,59,61,67,71,73,79,83,89,97.

Multiples of 2

Multiples of 2

		2	3	4	5	6	7	8	9	10
10	11	12	13	14	15	16	17	18	19	20
20	21	22	23	24	25	26	27	28	29	30
30	31	32	33	34	35	36	37	38	39	40
40	41	42	43	44	45	46	47	48	49	50
										100

Multiples of 3

Multiples of 3

		2	3	-	5	-	7	-	9	-
-	11	-	13	-	15	-	17	-	19	-
-	21	-	23	-	25	-	27	-	29	-
-	31	-	33	-	35	-	37	-	39	-
-	41	-	43	-	45	-	47	-	49	-
										-

Multiples of 5

Multiples of 5

		2	3	-	5	-	7	-	-	-
-	11	-	13	-	-	-	17	-	19	-
-	-	-	23	-	25	-		-	29	-
-	31	-	-	-	35	-	37	-	-	-
-	41	-	43	-	45	-	47	-	49	-
										-

Multiples of 7

Multiples of 7

		2	3	-	5	-	7	-	-	-
-	11	-	13	-	-	-	17	-	19	-
-	-	-	23	-	-	-		-	29	-
-	31	-	-	-	-	-	37	-	-	-
-	41	-	43	-	-	-	47	-	49	-
										-

Primes Less than 100

Primes Less than 100

		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			-

- Generating primes is a practical very problem.
- A large class of digital security and Cryptography algorithm rely on prime numbers.
- However, there is no known closed formula or function which always produces primes
- The function $f(n) = n^2 - n + 41$ results in prime numbers for all choices of n between 1 and 40. However, $f(41) = 41^2$ is not prime

Theorem 2.2

There are infinitely many prime numbers

- Proof the above theorem

- Though there are infinitely many primes, generating them is a challenge.
- The trial division or the sieve of eratosthenes, can determine if a number is prime.
- However for large numbers, these methods are inefficient.
- Another test of primality is based on **Fermat's little theorem**

Theorem 2.3: Fermat's little theorem

For two possitive integers a and p if p is prime and $p \nmid a$, then $a^{p-1} \equiv 1 \bmod p$

- Since 5 is prime and $5 \nmid 16$, By Fermat's little theorem, it must be the case that $16^4 \equiv 1 \bmod 5$.
- Indeed we have $16^4 = 2^{4^4} = 2^{16} = 65536$ and $65536 \equiv 1 \bmod 5$
- Fermat's little theorem gives rise to the Fermat primality test.

Probablilistic Method

- For any +integer n, we can deduce that n is not prime if we can find a number a such that $a^{n-1} \not\equiv 1 \bmod n$. Such an a is called Fermat witness
- Fermat probability test leads to a probablilistic method to determine if a number is prime.
- Probabilistic algorithm is one which produces the correct result "with high probability" but not necessarily all the time.
- It tries to find Fermat witness. If after a certain number of attempts it cannot find such witness, then the algorithm terminates and assumes that the number is prime.
- The assumption is what makes the algorithm probabilistic. The number of times to perform the test is not clear.

- A python code implementing the Fermat Primality test.
- A random number *a* is chosen as a possible Fermat Witness. It generates random primes by choosing using this test

```
from random import randint

def isPrime(p,numInter):
        for i in range(numInter):
                a=randint(2,p-1)
                e=a**(p-1)%p
                if $(e!=1):
                        return False:
                return True

def randomPrime(n):
        while(True):
                p=randint(2**(n-1), 2**(n)-1):
                if isPrime(p,128)
                        return p;
 #print a random 32-bit prime
 print(randomPrime(16))

 output: 49103
```

Prime Conjectures

- Primes have been studied for thousand of years by countless researchers yet many properties are unproved
- **Goldbach's conjecture** Every even integer n greater than 2 is the sum of two primes. This has been verified for numbers up to 1.6×10^{18}
- **Landau's conjecture** There are infinitely many primes of the form $n^2 + 1$ for a positive integer n
- **Twin Prime conjecture** There are infinitely many primes that differ by 2. Twin prime include 5 and 7, 11 and 13, 71 and 73 e.t.c.

Definition 2.2: GCD

For two non-zero integers a and b, d is the greatest common divisor of a and b if $d|a, d|b$, and any other common divisor of a and b also divides d

Definition 2.3: Co-prime

Two integers are relatively prime if their Greatest Common Divisor is 1. Such integers are co-prime

- The GCD of two numbers can be determined if they are relatively prime based on their factorization

$$a = p_1^{e_1} p_2^{e_2} \cdots p_n^{e_n}$$
$$b = q_1^{e_1} q_2^{e_2} \cdots q_n^{e_n}$$

- If any of the primes p_i equals a prime p_j, then a and b have a non trivial GCD.

- If a and b have no primes in common between their prime factorization, then they are relatively prime
- The GCD of a and b can be computed from their prime factorization by getting the product of all common prime raised to the minimum exponent of that prime in either number.

Example 2.1: GCD from Primes

To compute the GCD of 1470 and 350 then

$$1470 = 2.3.5.7^2$$
$$350 = 2.5^2.7$$

The GCD of 1470 and 350 is thus $2^{\min(1,1)}.5^{\min(1,2)}.7^{\min(2,1)} = 70$

Least Common Multiples

- Prime factorization can also used to compute the LCM between two numbers.

Definition 2.4: Least Common Multiple

The LCM of two positive integers *a* and *b* is the smallest positive integer that is divisible by both *a* and *b*

- GCD is computed by taking the common primes of two integers(i.e. the intersection of primes in their factorization) whereas the LCM is computed by taking the Union of primes in their factorization. In this case, each prime is raised to the maximum exponent of that prime in either number

Example 2.2: Computing LCM from primes

We compute the LCM of 1470 and 350 thus

$$1470 = 2.3.5.7^2$$
$$350 = 2.5^2.7$$

The LCM of 1470 and 350 is thus $2^{\max(1,1)}.5^{\max(1,2)}.7^{\max(2,1)} = 7350$

Least Common Multiples

Computing the prime factorization of a number in general is a challenge. Therefore the above method for computing the LCM is not practical. A more practical method is the Euclidean Algorithm

Theorem 2.4

For any two positive integers a and b, we have

$$a \cdot b = \gcd(a, b) \cdot \operatorname{lcm}(a, b)$$

- This Algorithm is efficient for computing the GCD of two numbers.
- Has been there for many years and it is attributed to Euclid.
- Based on simple idea rooted in Euclidean division.

Let $a = bq + r$ by Euclidean division.
Then, by rearranging $r = a - bq$
If we let d be a common divisor of a and b,
Thus $d \mid a$ and $d \mid b$, Certainly, then we have $d \mid r$ since $r = a - bq$

Lemma 2.1

Let a and b be integers with $a = bq + r$ by Euclidean division, Thus $0 \leq r < b$

$$\gcd(a, b) = \gcd(b, r)$$

Example 2.3: Euclidean by example

Let us find the GCD of 287 and 91 by the Euclidean algorithm.

$$287 = 91 \cdot 3 + 14 : \ 287 \text{ mod } 91 = 14$$

$$91 = 14 \cdot 6 + 7 : \ 91 \text{ mod } 14 = 7$$

$$14 = 7 \cdot 2 + 0 : \ 14 \text{ mod } 7 = 0$$

Since we cannot divide by 0 in the next step, the process terminates and we have:

$$\text{gcd}(287, 91) = \text{gcd}(91, 14) = \text{gcd}(14, 7) = \text{gcd}(7, 0) = 7$$

In this example, notice remainder sequence starting at 287 and ending at 0

$$
\begin{aligned}
r_0 &= 287 \\
r_1 &= 91 \\
r_2 &= 14 \\
r_3 &= 7 \\
r_4 &= 0
\end{aligned}
$$

- Notice that the *remainder sequence* is strictly decreasing.
- From the Euclidean division(ED) we have $a = bq + r$ with $0 \leq r < b$. Since the magnitude of the remainder strictly reduces, and has a lower 0 as lower bound, then repeated ED will lead to a 0 remainder, terminating the algorithm. The correctness follows from the previous lemma.

Algorithm gcd(a,b)

```
Algorithm gcd(a,b)
Input: $a, b \in \mathbb{Z}^+, a > b$
Output $x$, the GCD of $a$ and $b$
    $x \longleftarrow a$
    $y \longleftarrow b$
     while $y \neq 0$ do
          $r \longleftarrow x$ mod $y$
          $x \longleftarrow y$
          $y \longleftarrow r$
     end while
return x
```

```
def gcd(a,b) :
x = a
y = b
print("r0: %d" % x)
print("r1: %d" % y)
i = 2;
while y != 0 :
r = x % y
print("r%d: %d" % (i, r))
i += 1
x = y
y = r
return x

print("GCD(152152, 154700) =
    % gcd(152152, 154700))
```

Ouput of the above program

```
r0: 152152
r1: 154700
r2: 152152
r3: 2548
r4: 1820
r5: 728
r6: 364
r7: 0
```

GCD(152152, 154700) = 364

Bezout Relations and GCDs

- The GCD have the property that it can be expressed as a linear combination of the two input integers

Theorem 2.5: Bezout Theorem

For any positive integers a and b there exists integers s and t such that $gcd(a, b) = sa + tb$

- The formula $gcd(a, b) = sa + tb$ is called *bezout identity* and the integers s and t are called the *Bezout coefficients* of a and b

Example 2.4

Find the be Bezout coefficient of 6 and 14

$$gcd(6, 14) = 2$$
$$gcd(6, 14) = (-2).6 + 1.14 = 2$$

The Bezout coefficients of 6 and 14 are -2 and 1

Bezout Relations and GCDs

- The Bezout coefficient of two numbers can be computed through "two pass" method using the Euclidean Algorithm. Consider the Euclidean algorithm for computing the GCD of 252 and 198.

$$\begin{aligned} 252 &= 1 \cdot 198 + 54 \\ 198 &= 3 \cdot 54 + 36 \\ 54 &= 1 \cdot 36 + 18 \\ 36 &= 2 \cdot 18 + 0 \end{aligned}$$

- The $gcd(252, 198) = 18$. After this we express 18 as a combination of 252 and 198 through successive bottom-up successive Euclidean divisions and back-substitution of one equation at time

$$\begin{aligned} 18 &= 54 - 1 \cdot 36 \\ 18 &= 54 - 1 \cdot (198 - 3 \cdot 54) = 4 \cdot 54 - 1 \cdot 198 \\ 18 &= 4 \cdot (252 - 1 \cdot 198) - 1 \cdot 198 = 4 \cdot 252 - 5 \cdot 198 \end{aligned}$$

Consequences of Bezout

Lemma 2.2

Let a, b, c be positive integers such that a and b are relatively prime. If $a \mid bc$ then $a \mid c$

Proof.

Since a and b are relatively prime then $gcd(a, b) = 1$
By Hypothesis, assume $a \mid bc$
By Bezout theorem, then $sa + tb = \text{gcd}(a, b) = 1$
Multiplying both side c

$$csa + ctb = c$$

Since

$$a \mid bc, a \mid ctb$$

That is there exists q such that $ctb = qa$

$$csa + ctb = c$$
$$csa + qa = c$$
$$a(cs + q) = c$$

Hence, $a \mid c$ as required □

Consequences of Bezout

Lemma 2.3

Let p be a prime integer and $a_1, a_2, \ldots, a_n$ be integers.
If $p \mid a_1 a_2 \cdots a_n$ then $p \mid a_i$ for at least one i

Though Division does not always maintain congruence relations as is the case for sums and products, it does under Bezout relations

Theorem 2.6

Let m be a positive integer and a, b, c be integers. If $gcd(c, m) = 1$ and $ac \equiv bc \bmod m$, then $a \equiv b \bmod m$

Proof.

By Hypothesis $ac \equiv bc \bmod m$ hence $m \mid ac - bc = c(a - b)$. And from the previous lemma, m must therefore divide c or $(a - b)$. Since, by assumption, $\gcd(c, m) = 1$, it must be that $m \mid a - b$. That is $a \equiv b \bmod m$ □

Exercises

1. Write a Java program with a public function that takes an integer n and prints all prime numbers between 2 and n. Use the sieve of Eratosthenes as your algorithm
2. Determine the following values
   - a. $\gcd(-24, 18)$
   - b. $\gcd(756, 210)$
   - c. $\gcd(-756, 210)$
   - d. $\gcd(742, 14)$
3. Compute the prime factorization of the following numbers
   1. 78
   2. 672
   3. 7920

Exercises

RSA is a cryptography system which relies on exponentiation and modula numbers. In particular, it is relatively easy to find three integers e, d, m such that, for any integer $n, 0 \leq n < m$ $(n^e)^d \equiv n \bmod m$. We call n^e the *encrypted message* and e the *public key*. Then $(n^e)^d \equiv n$ is the *decrypted message* and d is the *private key*

In particular, we can compute such an m as the *least common multiple* of $p - 1$ and $q - 1$ for two different prime numbers p and q. If $p = 7$ and $q = 11$ then $m = 30$

Find e and d such that $(3^e)^d \equiv 3 \bmod 30$

In the previous section we saw positive integer forms a special kind of equivalence known as congruence relation of the form $4 \equiv 16 \bmod 6$ since $6 \mid 16 - 4$. In this section variables are included to make the equations comprehensive.

A linear congruence is an equivalence of the form $ax \equiv b \bmod m$ where x is a variable, a, b are positive integers and m is the modulus.

- The solution to this congruence is all integers x which satisfy the congruence

Example 3.1

$$2x \equiv 1 \bmod 5$$

By inspection we find: $2 \cdot 3 = 6 \equiv 1 \bmod 5$

Solution to this congruence has $x = 3$

However, notice that $x = 8$ is also a solution as $2 \cdot 8 = 16 \equiv 1 \bmod 5$

- Linear congruence have infinitely many possible solutions.
- In the above example since $x = 3$ was a solution, then so is every element in the congruence class of 3. Recall $\overline{3} = \{x \mid x \equiv 3 \mod 5\}$.

Modular Inverses

Solving the equation of the form $ax = b$ over the reals.

- Normally divide through by a assuming $a \neq 0$ to get $x = \frac{b}{a}$
- This is equivalent to multiplying both sides by the *multiplicative inverse of a* - *This is another number such that their product is identity.*
- Over real numbers $(\mathbb{R})$ then $a \cdot \frac{1}{a} = 1$ for any a
- As is the case over rational numbers or real numbers, there are (often, but not always) multiplicative inverses when working modulo a number.
- Given a number x and a modulus m, the multiplicative inverse of x is another number a such that $ax \equiv 1 \mod m$

Modular Inverses

Example 3.2

Compute the inverse of 3 modulo 7

$$3a \equiv 1 \mod 7$$
$$15 \equiv 1 \mod 7$$
$$\rightarrow a \equiv 5 \mod 7$$

- Since in the above example, 5 is the modular inverse 3 mod 7, any number in the congruence class of 5 modulo is a multiplicative inverse.
- Modulo inverses can be used to solve linear congruences. Let a' be the inverse of a modulo m, then we have the following relations

$$ax \equiv b \mod m$$
$$a'ax \equiv a'bx \mod m$$
$$x \equiv a'b \mod m$$

Modular Inverses May Not Exist

Sometimes some certain numbers do not have multiplicative inverses modulo a particular modulus.

Example 3.3

Consider 2 modulo 6. It does not have an inverse. This can be verified by multiplying 2 by each of the $\{0, 1, 2, 3, 4, 5\}$

$$\begin{aligned}
2 \cdot 0 &\equiv 0 \mod 6 \\
2 \cdot 1 &\equiv 2 \mod 6 \\
2 \cdot 2 &\equiv 4 \mod 6 \\
2 \cdot 3 &\equiv 0 \mod 6 \\
2 \cdot 4 &\equiv 2 \mod 6 \\
2 \cdot 5 &\equiv 4 \mod 6
\end{aligned}$$

But when do they not exist? This is because there is no identity and also $gcd(2, 6) \neq 1$ and also the fact that, 2 is called the a zero-divisor for arithmetic modulo m

Computing Modular Inverses

When the number is small, inspection comes in handy to compute the inverse. eg. What is the inverse of 3 mod 8?

- What is the inverse of 151 mod 951

Theorem 3.1

If a and m are relatively prime integers with $m > 1$, then there exists a unique modular inverse x of a mod m satisfying $0 < x < m$

Proof

Proof.

First we proof the existence of x.
By Hypothesis we have $gcd(a, m) = 1$. Therefore by Bezout theorem there exists integers s and t such that $sa + tm = 1$
We therefore have:

$$1 - sa = tm$$

$$m \mid (1 - sa)$$

$$1 \equiv sa \bmod m$$

$\therefore s = x$ is the modular inverse of m

Next, we show uniqueness.
Assume that there is another modular inverse b of a
By definition of modular inverses we have $xa \equiv 1 \bmod m$ and $ba \equiv 1 \bmod m \therefore xa \equiv ba \bmod m$. From a previous theorem $x \equiv b \bmod m$ since $\gcd(a, m) = 1$
$\therefore x$ is unique for $0 < x < m$ □

Examples

Example 3.4: Easy Example

Find the inverse of 3 modulo 7.
The $gcd(3,7) = 1$, $\therefore$ an inverse must exist. From Euclidean Division we have $7 = 2 \cdot 3 + 1$ and thus $-2 \cdot 3 + 1 \cdot 7 = 1$. Hence -2 is the Bezout coefficient of 3 and $-2 \equiv 5 \bmod 7$ is the modular inverse of 3

Extended Modular Inverse

Find the inverse of 151 modulo 951

Using the Euclidean Algorithm, we find

$$951 = 6 \cdot 151 + 45$$
$$151 = 3 \cdot 45 + 16$$
$$45 = 2 \cdot 16 + 13$$
$$16 = 1 \cdot 13 + 3$$
$$13 = 4 \cdot 3 + 1$$
$$3 = 3 \cdot 1 + 0$$

$\therefore, \gcd(951, 151) = 1$

The Bezout relation between 951 and 151 via back substitution

$$1 = 13 - 4 \cdot 3$$
$$1 = 13 - 4(16 - 1 \cdot 13) = -4 \cdot 16 + 5 \cdot 13$$
$$1 = -4 \cdot 16 + 5(45 - 2 \cdot 16) = 5 \cdot 45 - 14 \cdot 16$$
$$1 = 5 \cdot 45 - 14(151 - 3 \cdot 45) = -14 \cdot 151 + 47 \cdot 45$$
$$1 = -14 \cdot 151 + 47(951 - 6 \cdot 151) = -296 \cdot 151 + 47 \cdot 951$$

$\therefore$ the modular inverse of 151 modulo 951 is $-296 \equiv 655 \bmod 951$

Example

Solve the following linear congruence by first computing an inverse.
$57x \equiv 13 \bmod 67$

Solution

First find the inverse of 57 modulo 67 . Use the Euclidean algorithm:

Solution

First find the inverse of 57 modulo 67 . Use the Euclidean algorithm:

$$67 = 1 \cdot 57 + 10$$

$$57 = 5 \cdot 10 + 7$$

$$10 = 1 \cdot 7 + 3$$

$$7 = 2 \cdot 3 + 1$$

$$3 = 3 \cdot 1 + 0$$

Solution

Since the $gcd(67, 57) = 1$ so a modular exists. Follow this then by computing the Bezout coeffiecients

Solution

Since the $gcd(67, 57) = 1$ so a modular exists. Follow this then by computing the Bezout coeffiecients

$$1 = 7 - 2 \cdot 3$$

$$1 = 7 - 2(10 - 1 \cdot 7) = -2 \cdot 10 + 3 \cdot 7$$

$$1 = -2 \cdot 10 + 3(57 - 5 \cdot 10) = 3 \cdot 57 - 17 \cdot 10$$

$$1 = 3 \cdot 57 - 17(67 - 1 \cdot 57) = -17 \cdot 67 + 20 \cdot 57$$

Solution

The inverse of 57 modulo 67 is 20. This yields

Solution

The inverse of 57 modulo 67 is 20. This yields

$$57x \equiv 13 \mod 67$$

$$20(57x) \equiv 20 \cdot 13 \mod 67$$

$$1 \cdot x \equiv 260 \mod 67$$

$$x \equiv 59 \mod 67$$

Systems of Linear Congruences

For a system of linear congruences of the form

$$\begin{cases} x \equiv 3 \mod 7 \\ x \equiv 6 \mod 13 \end{cases}$$

Can we find a value of x that simultaneously satisfies both of these equations? Inspecting, 45 is a possible solution as $7 \mid (45 - 3) = 42$ and $13 \mid (45 - 6) = 39$

Is there an algorithmic process to find such an x? Yes

Chinese Remainder Theorem

Theorem 3.2: Chinese Remainder Theorem

Let m, n be two co-prime integers greater than 1 then

$$\begin{cases} x \equiv a \mod m \\ x \equiv b \mod n \end{cases}$$

Has a unique solution modulo $m \cdot n$

Chinese Remainder Theorem - Proof

Proof.

Since m and n are co-prime then by bezout theorem, there exists integers s, t such that;

$$sm + tn = 1$$

Then notice that $x = bsm + atn$ satisfies the linear congruences

$$\begin{aligned} x &= bsm + atn \\ &= bsm + a(1 - sm) \\ &= bsm + a - asm \\ &\equiv a \mod m \end{aligned}$$

$$\begin{aligned} x &= bsm + atn \\ &= b(1 - tn) + atn \\ &= b - btn + atn \\ &\equiv b \mod n \end{aligned}$$

Consider uniqueness. Let $x = y$ and $x = z$ be two solutions of this system. Then, y and z must give the same remainder when divided by m or n. $\therefore m \mid y - z$ and $n \mid y - z$. Since m and n are co-prime, it follows that $m \cdot n \mid y - z \therefore y \equiv z \bmod m \cdot n$ □

CRT - Examples

Example 3.5: Linear Congruences

Find all integers $x, 0 \leq x < 15$ such that

$$\begin{cases} x \equiv 1 \mod 3 \\ x \equiv 2 \mod 5 \end{cases}$$

Since 3 and 5 are co-prime, CRT states that there exists a unique solution modulo 15

$\therefore$ exactly one solution x with $0 \leq x < 15$

Applying the Euclidean algorithm to find s, t such that $3s + 5t = 1$

Or, by inspection $3(2) + 5(-1) = 1 \therefore s = 2, t = -1$

Thus $x = 2(3s) + 1(5t) = 2(3)(2) + 1(5)(-1) = 7 \equiv 7 \bmod 15$

CRT - Examples

Example 3.6: Linear Congruences

For the following Linear congruences solve for x where $0 \leq x < 91$

$$\begin{cases} x \equiv 3 \mod 7 \\ x \equiv 6 \mod 13 \end{cases}$$

Using Euclidean Algorithm 7 and 13 are co-prime

$\therefore$ there exist s, t such that $7s + 13t = 1$

By inspection $s = 2$ and $t = -1$

$\therefore$ a solution $x = 6(7s) + 3(13t) = 84 - 39 = 45$

Hence we verify that $7 \mid 45 - 3 = 42$ and $13 \mod 45 - 6 = 39$

1. Find the Multiplicative inverse of:
   a. 8 modulo 17
   b. 9 modulo 13
   c. 11 modulo 71
2. Prove the follwing Lemma

Lemma 3.1

Let m and n be co-prime integers
For any integer x such that $m \mid x$ and $n \mid x$ then $mn \mid x$

Solution of the Lemma in the exercises

Proof.

By Hypothesis $m \mid x$ and $n \mid x$

$\therefore, \exists\ q_m, q_n$ such that $x = mq_m$ and $x = nq_n$ hence $mq_m = nq_n$

Since m and n are co-primes, $\exists\ s, t$ such that $sm + tn = 1$

$\therefore, smq_m + tnq_m = q_m$ and combining with our hypothesis gives

$snq_n + tnq_m = q_m$ and thus

$n(sq_n + tq_m) = q_m$

From $x = mq_m$ we get

$x = mn(sq_n + tq_m)$ Hence

$mn \mid x$ □

Excises Cont'd

3. Solve the following linear congruences. Give the unique positive solution which is less than the modulus.
   1. $x \equiv 12 \mod 7$
   2. $2x \equiv 12 \mod 7$
   3. $13x \equiv 15 \mod 23$
4. Solve the following system of linear congruences for $0 \leq x < 77$

$$\begin{cases} 3x \equiv 2 \mod 11 \\ 4x \equiv 6 \mod 7 \end{cases}$$

5. Solve the following system of linear congruences for $0 \leq x < 221$

$$\begin{cases} 4x \equiv 11 \mod 13 \\ 2x \equiv 7 \mod 17 \end{cases}$$

Consider the number 496.

This number, Four Hundred and Ninety Six means what?

Rarely do we think about about this because the *decimal system* of numbers was introduced to us from the beginning.

However, there are many ways we could represent and understand numbers

Positional Number System

- Other than using roman numeral and or series of dots to encode the things being counted it is much more convenient to use a *positional number system*
- In the roman numerals *I* mean 1 *v* means $5\cdots$, the position of the symbol does not change its value
- Decimal representation of number is positional. Putting two 1 symbols together to form 11 means something different from *II* in Roman numerals.
- Decimal System of numbers has *radix* or *base* 10.
- Position of a digit in a number represents a multiple of a certain power of 10

$$12345 = (1 \cdot 10^4) + (2 \cdot 10^3) + (3 \cdot 10^2) + (4 \cdot 10^1) + (5 \cdot 10^0)$$

Radix-r Representations

The positional number system can be represented with any choice of radix.

In modtern times base 10 has been used as the main representation because humans have 10 fingers.

Historically, different groups used different bases. The Mayans used base 20 (10 toes and 10 fingers). The Babylonians used base 60. This is why time and angles are measured in groups of 60 seconds, 60 minutes.

Given some radix r, we can construct a number system using r as the radix or base. This results from the following theorem.

Theorem 4.1

Let r be a positive integer greater than 1. Any positive integer n can be expressed uniquely in the form:

$$n = a_k r^k + a_{k-1} r^{k-1} + \cdots + a_2 r^2 + a_1 r + a_0$$

where k is a non-negative integer, $a_j (0 \leq j \leq k)$ belongs to the set

$\{0, 1, \ldots, r-1\}$ and $a_k \neq 0$

This formula for n is called the *radix-r representation of n*.

Radix-*r* Representations

In the modern technological age, base-2, base-8, and base-16 are important number systems.

1. Base-2 (binary) is used throughout electronics as the “digital numbers”. Each digit is 0 or 1, a bit, representing “off” or “on” of the electrical voltage.
2. Base-8 (octal) is used throughout computing where numbers were represented using $6, 12$, or 24 bits, and thus $2, 4$, or 8 octal digits.
3. Base-16 (hexadecimal) has become popular in computing where computers now represent numbers using 32 or 64 bits or and thus 8 hex digits.

Radix-r Representations

Example 4.1: Binary Numbers

$$(1010)_2 = (1 \cdot 2^3) + (0 \cdot 2^2) + (1 \cdot 2^1) + (0 \cdot 2^0) = 8 + 2 = 10$$

$$(10101)_2 = (1 \cdot 2^4) + (0 \cdot 2^3) + (1 \cdot 2^2) + (0 \cdot 2^1) + (1 \cdot 2^0) = 16 + 4 + 1 = 21$$

$$(1111111111111111)_2 = \sum_{i=0}^{15} 2^i = 2^{16} - 1 = 65535$$

A binary number with n digits has a value which ranges from 0 to $2^n - 1$

Radix-r Representations

Hexadecimal is the most obscure of the number system because of use of letters as numbers (rather than variables representing numbers). In hexadecimal, $A = 10, B = 11, C = 12, D = 13, E = 14$ and $F = 15$

Example 4.2: Hexadecimal numbers

$$(123)_{16} = (1 \cdot 16^2) + (1 \cdot 16^1) + (1 \cdot 16^0) = 256 + 2(16) + 3 = 291$$

$$\begin{aligned}(BC123)_{16} &= (11 \cdot 16^4) + (12 \cdot 16^3) + (1 \cdot 16^2) + (2 \cdot 16^1) + (3 \cdot 16^1)\\ &= 11(65536) + 12(4096) + 291\\ &= 770339\end{aligned}$$

Radix-r Representations - Exercises

Convert the below numbers to decimal numbers.

1. $(11001)_2$
2. $(1000011001)_2$
3. $(7612)_8$
4. $(7612)_{16}$

Radix-r numbers in practice

In practice, we write radix-r representations different from the mathematical way of $(23DFEA4)_{16}$

- *Binary numbers* "0b" prefix `0b101101` $= 45$
- *Octal Numbers* "0" or "0o" prefix `0o12654` $= 5548$
- *Hexadecimal Numbers* "0x" prefix `0x23DFEA4` $= 37617316$

•Programming languages feature native support for many different radix representations.

• For example in Python one can define a "literal" numbers in different radix representations use the aforementioned prefixes.

• Convert a decimal number to binary with `bin()`, to octal with `oct()`, and to hexadecimal with `hex()`.

• Note that these functions return a string representing the number.

Converting to radix-r

- Radix-r representations are an expansion of a number using powers of as the base. This suggests that converting from decimal to a radix- r representation can be performed by (repeated) division.
- Let r be some radix and n be some integer number to convert to radix- r. By Euclidean division we have:

  $$n = q_0 r + a_0$$

  with $0 \leq a_0 < r$
- Notice that a_0 is thus a digit in the radix-r number system. In fact, a_0 is the first digit (counting from the right) of the radix-r representation or n.
- Continue dividing q_0 by r

  $$q_0 = q_1 r + a_1$$

  with $0 \leq a_1 < r$ Again, a_1 is a digit in the radix-r number system and a_1 is the second digit of the radix-r representation of n.
- This continues until a q_k is 0, using the successive remainders as the digits of the radix-r representation of n

Converting to radix-r

Algorithm radix_r_expansion(n, r)

```
**Input:** $n, r \in \mathbb{Z}^+,\ r > 1$
**Output:** base $r$ expansion of n: $(a_{k-1} \cdots a_1 a_0)_r$.
$q \leftarrow n$
$k \leftarrow 0$
**while** $q \neq 0$ **do**
    $a_k \leftarrow q \mod r$
    $q \leftarrow q$ **div** $r$
    $k \leftarrow k + 1$
**end while**
**return** $(a_{k-1} \cdots a_1 a_0)$
```

Converting to radix-r

Example 4.3: Converting to Radix-16

Convert 93752 to hexadecimal

$$
\begin{aligned}
93752 &= 5859(16) + 8 \\
5859 &= 366(16) + 3 \\
366 &= 22(16) + 14 \\
22 &= 1(16) + 6 \\
1 &= 0(16) + 1
\end{aligned}
$$

$\therefore 93752 = (16E38)_{16}$

Converting to radix-r

How does this work?

Notice that we have $n - a_0 = q_0 r$.

Hence, $r \mid n - a_0$

and $n \equiv a_0 \bmod r$

This makes sense because we want:

$$\begin{aligned} n = a_k r^k + a_{k-1} r^{k-1} + \cdots + a_2 r^2 + a_1 r + a_0 &\leftrightarrow n \equiv a_0 \bmod r \\ &\leftrightarrow n \equiv a_1 r + a_0 \bmod r^2 \\ &\vdots \end{aligned}$$

Binary, Octal, Hex Conversion

- Recall that binary, octal, and hexadecimal numbers all have roots in electronics and computing.
- Each of these representations have their own merits in computer science.
- Therefore very useful to be able to convert between these representations. Doing such a conversion is very easy.
- The binary system uses one digit to represent each bit in a computer system.
- The octal system, with digits 0 through 7 represents three binary digits at once: $7 = (111)_2, 6 = (110)_2$, etc.
- Therefore, we can easily convert from binary to octal by grouping binary digits into threes and then converting each group to its corresponding decimal (octal) digit.

Example 4.4: Binary to Octal

$$(11010101000101010010 10)_2 \rightarrow 001\ \ 101\ \ 010\ \ 100\ \ 010\ \ 101\ \ 001\ \ 010$$

$$\rightarrow 1\ \ 5\ \ 2\ \ 4\ \ 2\ \ 5\ \ 1\ \ 2$$

$$\rightarrow (15242512)_8$$

Hexadecimal system represents 4 binary digits at once $15 = (1111)_2, 14 = (1110)_2 \cdots$

Example 4.5: Binary to Hexadecimal

$$(1101010100010101001010)_2 \rightarrow 0011\ \ 0101\ \ 0100\ \ 0101\ \ 0100\ \ 1010$$

$$\rightarrow 3\ \ 5\ \ 4\ \ 5\ \ 4\ \ A$$

$$\rightarrow (35454A)_{16}$$

Binary Arithmetic

- Regardles of which number systems, arithmetic like addition, multiplication, division, etc. always give the same result.
- Whether in binary, hexadecimal, octal, or decimal, the sum of two numbers is still its sum.
- The only thing that changes is the way we write down the numbers being added and their sum.
- Doing addition and subtraction in the binary number system is not so different from doing it in the decimal system.
- The key is to understand how we add individual digits, just like in “long addition” in the decimal system.
- There are three possibilities for adding single bits: both are 0, both are 1, or one is 1 and the other is 0

$$(0)_2 + (0)_2 = (0)_2$$

$$(1)_2 + (0)_2 = (1)_2$$

$$(1)_2 + (1)_2 = (10)_2$$

There are four cases for adding 3 bits together: there are zero 1s, there is one 1, there are two 1s, there are three 1s.

$$(0)_2 + (0)_2 + (0)_2 = (0)_2$$
$$(1)_2 + (0)_2 + (0)_2 = (1)_2$$
$$(1)_2 + (1)_2 + (0)_2 = (10)_2$$
$$(1)_2 + (1)_2 + (1)_2 = (11)_2$$

Using this basic addition of three bits and the ideas of “carrying” digits, we can compute the addition of any two binary numbers.

Example 4.6: Binary Addition

Compute $(1101)_2 + (110)_2$ in binary

Binary Multiplication

Binary multiplication can be derived from binary addition with a simple observation. Let $m = (a_k a_{k-1} \cdots a_2 a_1)_2$ and $n = (b_\ell b_{\ell-1} \cdots b_2 b_1)_2$

Then

$$m \cdot n = m \cdot (b_\ell 2^\ell + b_{\ell-1} 2^{\ell-1} + \cdots + b_2 2^2 + b_1 2 + b_0)$$

$$= m b_\ell 2^\ell + m b_{\ell-1} 2^{\ell-1} + \cdots + m b_2 2^2 + m b_1 2 + m b_0$$

As m is itself a binary number each term $m b_\ell 2^\ell$ has a simple computation

First, since b_ℓ is a binary digit, it is either 0 or 1.

If it is 0, the product is also 0.

If it is 1 then $a b_\ell 2^\ell = a 2^\ell$

Notice

$$m 2^\ell = 2^\ell (a_k 2^k + a_{k-1} 2^{k-1} + \cdots + a_2 2^2 + a_1 2 + a_0)$$

$$= a_k 2^{k+\ell} + a_{k-1} 2^{k-1+\ell} + \cdots + a_2 2^{2+\ell} + a_1 2^{1+\ell} + a_0 2^\ell$$

Therefore, $m 2^\ell$ is just a shift

Binary Multiplication

The algorithm of computing the binary multiplication for two numbers m given as $m = (a_k a_{k-1} \cdots a_2 a_1)_2$ and n given as $n = (b_\ell b_{\ell-1} \cdots b_2 b_1)_2$ is as follows

1. Let $p = 0$
2. For $i = 0, \ldots, \ell$, if $b_i = 1$, then $p = p + (m \cdot 2^i)$
3. p is the product $m \cdot n$

Binary Shift

When multiplying a binary number by a power of 2, the result is simply a shift of digits to the left, with the corresponding number of 0 digits added on the right.

$(1101011)_2 \cdot 2^5 = 110101100000$

Example 4.7: Binary Multiplication

Compute the product of $(1010)_2$ and $(11001)_2$

In this case, we have two 1 digits correspoding to 2^3 and 2^1

Hence the product p is

$$\begin{aligned} p &= (11001)_2 2^3 + (11001)_2 2^1 \\ &= (11001000)_2 + (110010)_2 \end{aligned}$$

$$\begin{array}{rl} 00000000 & \text{(carry bits)} \\ 11001000 & \\ +\ 00110010 & \\ \hline (11111010)_2 & \end{array}$$

Exercises

1. Convert the following numbers to decimal
   a. $(11100)_2$
   b. $(11101101)_2$
   c. $(235014)_8$
   d. $(56D9A0D)_{16}$
2. Convert the following to Octal
   a. 174
   b. $(11100)_2$
   c. $(FFDE)_{16}$
   d. 262144
3. Convert 12847 to
   a. radix- 6 representation
   b. radix- 13 representation HINT Use digits $\{0, 1, \cdots 9, A, B, C\}$

Exercises

4. Write a Java function convert(n,r) which returns a string-encoding of the radix-r representation of the integer n. Assume that n is non-negative and that $1 < r \leq 10$ Therefore, you do not have to worry about digits like $A, B, C \cdots$
5. Using "long" binary multiplication, compute the product of $(101011)_2$ and $(1101)_2$