CS220/Math320 - Applied Discrete Mathematics

Integers

April 2, 2020

Integer Properties

- Integers are a natural component of everyday life and easy to understand.
- Number theory was studied primarily for its own sake for the better part of the last several thousand years, without any particular application as the goal.
- In the last few decades number theory has emerged as a critical component of many applications, especially in computer science.
- In particular, number theory forms the mathematical basis for modern cryptography, the study of secure communication.

Integer Division

- If a and b are integers with $a \neq 0$, we say that a *divides* b if there is an integer c so that $b = ac$.
- When a divides b we say that a is a factor of b and that b is a multiple of a.
- The notation $a|b$ means that a divides b.
- We write $a \nmid b$ when a does not divide b.
- For integers a, b, and c it is true that
  - if $a|b$ and $a|c$, then $a|(b+c)$
    Example: 3|6 and 3|9, so 3|15.
  - if $a|b$, then $a|bc$ for all integers c
    Example: 5|10, so 5|20, 5|30, 5|40, ...
  - if $a|b$ and $b|c$, then $a|c$
    Example: 4|8 and 8|24, so 4|24.

Prime Numbers

- A positive integer p greater than 1 is called *prime* if the only positive factors of p are 1 and p.
- A positive integer that is greater than 1 and is not prime is called *composite*.
- The fundamental theorem of arithmetic:
- Every positive integer can be written *uniquely* as the product of primes, where the prime factors are written in order of increasing size.
- **Examples:**

  $15 = 3 * 5$

  $48 = 2 * 2 * 2 * 2 * 3 = 2^4 * 3$

  $17 = 17$

  $100 = 2 * 2 * 5 * 5 = 2^2 * 5^2$

  $515 = 5 * 103$

Prime Numbers

- If n is a composite integer, then n has a prime divisor less than or equal to $\sqrt{n}$.
- This is easy to see: if n is a composite integer, it must have two divisors p_1 and p_2 such that $p_1 * p_2 = n$ and $p1 \geq 2$ and $p2 \geq 2$.
- p_1 and p_2 cannot both be greater than $\sqrt{n}$ because then $p1 * p2 > n$
- If the smaller number of p_1 and p_2 is not a prime itself, then it can be broken up into prime factors that are smaller than itself but ≥ 2.

The Division Algorithm

- Let a be an integer and d a positive integer.
- Then there are unique integers q and r, with $0 \leq r < d$, such that $a = dq + r$.
- In the above equation:
  - d is called the divisor,
  - a is called the dividend,
  - q is called the quotient, and
  - r is called the remainder.

The Division Algorithm

- **Example:** When we divide 17 by 5, we have $17 = 5 * 3 + 2$.
- 17 is the dividend,
- 5 is the divisor,
- 3 is called the quotient, and
- 2 is called the remainder.

The Division Algorithm

- **Another Example:** What happens when we divide -11 by 3?
- Note that the remainder cannot be negative.
- $-11 = 3 * (-4) + 1$.
- -11 is the dividend,
- 3 is the divisor,
- -4 is called the quotient, and
- 1 is called the remainder.

Common Divisors

- Let a and b be integers, not both zero.
- The largest integer d such that $d|a$ and $d|b$ is called the greatest common divisor of a and b.
- The greatest common divisor of a and b is denoted by gcd(a, b).
- **Example 1:** What is gcd(48, 72) ?
- The positive common divisors of 48 and 72 are: 1, 2, 3, 4, 6, 8, 12, 16, and 24, so gcd(48, 72) $= 24$.
- **Example 2:** What is gcd(19, 72) ?
- The only positive common divisor of 19 and 72 is 1, so gcd(19, 72) $= 1$.

Greatest Common Divisors

- Using prime factorizations:
  $a = p_1^{a_1} p_2^{a_2} \dots p_n^{a_n}$, $b = p_1^{b_1} p_2^{b_2} \dots p_n^{b_n}$,
- where $p_1 < p_2 < \dots < p_n$ and $a_i, b_i \in N$ for $1 \leq i \leq n$
- $gcd(a, b) = p_1^{\min(a_1, b_1)} p_2^{\min(a_2, b_2)} \dots p_n^{\min(a_n, b_n)}$
- **Example:**

$$a = 60 \qquad = 2^2 * 3^1 * 5^1$$
$$b = 54 \qquad = 2^1 * 3^3 * 5^0$$

- $gcd = 2^1 * 3^1 = 6$

Relatively Prime Integers

- **Definition:** Two integers a and b are *relatively prime* if gcd(a, b) = 1.
- **Examples:**

  Are 15 and 28 relatively prime?

  Are 55 and 28 relatively prime?

  Are 35 and 28 relatively prime?

Relatively Prime Integers

- **Definition:** Two integers a and b are *relatively prime* if gcd(a, b) $= 1$.
- **Examples:**

  Are 15 and 28 relatively prime?

  Are 55 and 28 relatively prime?

  Are 35 and 28 relatively prime?

  Yes, yes, no.

Relatively Prime Integers

- **Definition:** The integers $a_1, a_2, \ldots, a_n$ are pairwise relatively prime if $gcd(a_i, a_j) = 1$ whenever $1 \leq i < j \leq n$.
- **Examples:**
  Are 15, 17, and 27 pairwise relatively prime?

Relatively Prime Integers

- **Definition:** The integers $a_1, a_2, \ldots, a_n$ are pairwise relatively prime if $gcd(a_i, a_j) = 1$ whenever $1 \leq i < j \leq n$.
- **Examples:**
  Are 15, 17, and 27 pairwise relatively prime?
  No, because gcd(15, 27) $= 3$.

Relatively Prime Integers

- **Definition:** The integers $a_1, a_2, \ldots, a_n$ are pairwise relatively prime if $gcd(a_i, a_j) = 1$ whenever $1 \leq i < j \leq n$.
- **Examples:**
  Are 15, 17, and 27 pairwise relatively prime?
  No, because gcd(15, 27) $= 3$.
- Are 15, 17, and 28 pairwise relatively prime?

Relatively Prime Integers

- **Definition:** The integers $a_1, a_2, \ldots, a_n$ are pairwise relatively prime if $gcd(a_i, a_j) = 1$ whenever $1 \leq i < j \leq n$.
- **Examples:**
  Are 15, 17, and 27 pairwise relatively prime?
  No, because $\gcd(15, 27) = 3$.
- Are 15, 17, and 28 pairwise relatively prime?
  Yes, because $\gcd(15, 17) = 1$, $\gcd(15, 28) = 1$ and $\gcd(17, 28) = 1$.

Least Common Multiple

- **Definition:**
  The *least common multiple* of the positive integers a and b is the smallest positive integer that is divisible by both a and b.
- We denote the least common multiple of a and b by lcm(a, b).
- **Examples:**
  lcm(3,7) = 21
  lcm(4,6) = 12
  lcm(5,10) = 10

GCD and LCM

- **Example from before:**
- $a = 60 = 2^2 * 3^1 * 5^1$
- $b = 54 = 2^1 * 3^3 * 5^0$
- $gcd(a, b) = 2^1 * 3^1 * 5^0 = 6$
- $lcm(a, b) = 2^2 * 3^3 * 5^1 = 540$
- As you see, $a * b = gcd(a, b) * lcm(a, b)$

Modular Arithmetics

- Let a be an integer and m be a positive integer.
- We denote by $a \mod m$ the remainder when a is divided by m.

  $9 \mod 4 = 1$

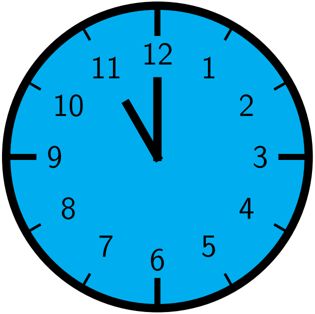
  $9 \mod 3 = 0$

  $9 \mod 10 = 9$

  $-13 \mod 4 = 3$

Modulo Congruence

- Let a and b be integers and m be a positive integer.
- We say that a is congruent to b modulo m if m divides a - b.
- We use the notation $a \equiv b(\mod m)$ to indicate that a is congruent to b modulo m.
- In other words: $a \equiv b(\mod m)$ if and only if $a \mod m = b \mod m$ (both leave the same remainder when divided by m).
- Everyday example of mod counting:



Examples:

- Is it true that $46 \equiv 68 \pmod{11}$?

Examples:

- Is it true that $46 \equiv 68 \pmod{11}$?

  Yes, because $11 | (46 - 68)$.
- Is it true that $46 \equiv 68 \pmod{22}$?

Modulo Congruence

Examples:

- Is it true that $46 \equiv 68 \pmod{11}$?

  Yes, because $11|(46-68)$.
- Is it true that $46 \equiv 68 \pmod{22}$?

  Yes, because $22|(46-68)$.
- For which integers z is it true that $z \equiv 12 \pmod{10}$?

Examples:

- Is it true that $46 \equiv 68 \pmod{11}$?

  Yes, because $11|(46-68)$.
- Is it true that $46 \equiv 68 \pmod{22}$?

  Yes, because $22|(46-68)$.
- For which integers z is it true that $z \equiv 12 \pmod{10}$?

  It is true for any $z \in \{\ldots, -28, -18, -8, 2, 12, 22, 32, \ldots\}$

Theorem

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

Modulo Congruence

Theorem

Let m be a positive integer. If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then $a + c \equiv b + d \pmod{m}$ and $ac \equiv bd \pmod{m}$.

Proof:

- We know that $a \equiv b$ (mod m) and $c \equiv d$ (mod m) implies that there are integers s and t with b = a + sm and d = c + tm.
- Therefore, $b + d = (a + sm) + (c + tm) = (a + c) + m(s + t)$ and $bd = (a + sm)(c + tm) = ac + m(at + cs + stm)$.
- Hence, $a + c \equiv b + d$ (mod m) and $ac \equiv bd$ (mod m)

The Euclidean Algorithm

- The Euclidean Algorithm finds the greatest common divisor of two integers a and b.
- It is based on the following lemma: if $a \equiv c(\mod b)$, then $gcd(a, b) = gcd(c, b)$.
- **Proof:** if $a \equiv c(\mod b)$, then $b|(a - c)$, so there is a y such that $a - c = by$, i.e., $c = a - by$.
- If any number d divides both a and b, then it also divides $a - by$.
- Therefore any common divisor of a and b is also a common divisor of c and b.
- Similarly, if d divides both c and b, then it also divides $c + by = a$, so any common divisor of c and b is a common divisor of a and b.
- This shows that the common divisors of a and b are exactly the common divisors of c and b, so, in particular, they have the same greatest common divisor.

The Euclidean Algorithm

- The Euclidean algorithm finds the smallest c in order to converge fast.
- For example, if we want to find gcd(287, 91), we divide 287 (the larger number) by 91 (the smaller one):

  287 = 91*3 + 14

  287 - 91*3 = 14

  287 + 91*(-3) = 14
- We know that for integers a, b and c, if $a|b$ and $a|c$, then $a|(b+c)$ for all integers c.
- Therefore, any divisor of 287 and 91 is also a divisor of 287 + 91*(-3), which is 14.
- Consequently, the gcd of 287 and 91 must be the same as the greatest common divisor of 14 and 91:

  gcd(287, 91) = gcd(91,14).

The Euclidean Algorithm

- In the next step, we divide 91 by 14: $91 = 14 * 6 + 7$
- This means that gcd(91, 14) = gcd(14, 7).
- So we divide 14 by 7: 14 = 7*2 + 0
- We find that $7|14$, and thus gcd(14, 7) = 7.
- Therefore, gcd(287, 91) = 7.

The Euclidean Algorithm

In pseudocode, the algorithm can be implemented as follows:

Algorithm 1 procedure gcd(a, b: positive integers)

```
x = a
y = b
while y ≠ 0 do
    r = x mod y
    x = y
    y = r
end while
return x
```

Representations of Integers

- Let b be a positive integer greater than 1 (the base).
- Then if n is a positive integer, it can be expressed uniquely in the form:
  $n = a_k b^k + a_{k-1} b^{k-1} + \cdots + a_1 b + a_0$,
  where k is a nonnegative integer, $a_0, a_1, \ldots, a_k$ are nonnegative integers less than b, and $a_k > 0$.
- Example for b=10:
  $859 = 8 * 10^2 + 5 * 10^1 + 9 * 10^0$
- Example for b=2 (binary expansion):
  $(10110)_2 = 1 * 2^4 + 1 * 2^2 + 1 * 2^1 = (22)_{10}$
- Example for b=16 (hexadecimal expansion):
  (we use letters A to F to indicate numbers 10 to 15)
  $(3A0F)_{16} = 3 * 16^3 + 10 * 16^2 + 15 * 16^0 = (14863)_{10}$

Representations of Integers

- How can we construct the base b expansion of an integer n?
- First, divide n by b to obtain a quotient q_0 and remainder a_0, that is,
  $n = bq_0 + a_0$, where $0 \leq a_0 < b$.
- The remainder a_0 is the rightmost digit in the base b expansion of n.
- Next, divide q_0 by b to obtain:
  $q_0 = bq_1 + a_1$, where $0 \leq a_1 < b$.
- a_1 is the second digit from the right in the base b expansion of n.
- Continue this process until you obtain a quotient equal to zero.

- **Example:** What is the base 8 expansion of $(12345)_{10}$?
- First, divide 12345 by 8:

  $12345 = 8 * 1543 + 1$

  $1543 = 8 * 192 + 7$

  $192 = 8 * 24 + 0$

  $24 = 8 * 3 + 0$

  $3 = 8 * 0 + 3$
- The result is: $(12345)_{10} = (30071)_8$.

Representations of Integers

Algorithm 2 base-b-expansion(n, b: positive integers)

q = n
k = 0
while $(q \neq 0)$ **do**
    $a_k = q \mod b$
    $q = \lfloor q/b \rfloor$
    $k = k + 1$
end while
return $(a_{k-1} \ldots a_1 a_0)$

Addition of Integers

How do we (humans) add two integers?

$$\begin{array}{r} {}^{1\,1\,1} \\ 7583 \\ +\,4932 \\ \hline 12515 \end{array}$$

Binary expansions:

$$\begin{array}{r} {}^{1\quad 1} \\ (1011)_2 \\ +\,(1010)_2 \\ \hline (10101)_2 \end{array}$$

Addition of Integers

- Let $a = (a_{n-1}a_{n-2}\ldots a_1a_0)_2$, $b = (b_{n-1}b_{n-2}\ldots b_1b_0)_2$.
- How can we algorithmically add these two binary numbers?
- First, add their rightmost bits:
  $a_0 + b_0 = c_0 * 2 + s_0$,
- where s_0 is the rightmost bit in the binary expansion of $a + b$, and c_0 is the carry.
- Then, add the next pair of bits and the carry:
  $a_1 + b_1 + c_0 = c_1 * 2 + s_1$,
- where s_1 is the next bit in the binary expansion of $a + b$, and c_1 is the carry.
- Continue this process until you obtain c_{n-1}.
- The leading bit of the sum is $s_n = c_{n-1}$.
- The result is: $a + b = (s_ns_{n-1}\ldots s_1s_0)_2$

Addition of Integers

- **Example:** Add $a = (1110)_2$ and $b = (1011)_2$.
- $a_0 + b_0 = 0 + 1 = 0 * 2 + 1$, so that $c_0 = 0$ and $s_0 = 1$.
- $a_1 + b_1 + c_0 = 1 + 1 + 0 = 1 * 2 + 0$, so $c_1 = 1$ and $s_1 = 0$.
- $a_2 + b_2 + c_1 = 1 + 0 + 1 = 1 * 2 + 0$, so $c_2 = 1$ and $s_2 = 0$.
- $a_3 + b_3 + c_2 = 1 + 1 + 1 = 1 * 2 + 1$, so $c_3 = 1$ and $s_3 = 1$.
- $s_4 = c_3 = 1$.
- Therefore, $s = a + b = (11001)_2$.

Addition of Binary Integers

Algorithm 3 add(a, b: positive integers)

c = 0
for j = 0 to n-1 {larger integer (a or b) has n digits} **do**
    $d = \lfloor (a_j + b_j + c)/2 \rfloor$
    $s_j = a_j + b_j + c - 2d$
    c = d
end for
$s_n = c$
return $(s_n s_{n-1} \ldots s_1 s_0)_2$