Mat 2345

Week 8

Fall 2013

Student Responsibilities — Week 8

- **Reading**: Textbook, Section 3.7, 4.1, & 5.2
- **Assignments**: Sections 3.6, 3.7, 4.1
  Induction Proof Worksheets
- **Attendance**: Strongly Encouraged

Week 8 Overview

- 3.6 Integers and Algorithms
- 3.7 Applications of Number Theory
- 4.1 Mathematical Induction

Section 3.6 — Integers and Algorithms

- **Euclidean Algorithm**: an efficient method of finding the greatest common divisor, rather than factoring both numbers.

- An example of how it works: `Find gcd(91,287)`

```
1. Divide the larger number by the smaller one:
   287 / 91 = 3 R 14, so
   287 = 91 (3) + 14
2. Any divisor of 91 and 287 must also be a
       divisor of 287 - 91(3) = 14
   Also, any divisor of 91 and 14 must also be a
       divisor of 287 = 91(3) + 14
3. Thus, gcd(91,287) = gcd(14,91); so divide 91 by 14
       91 = 14(6) + 7
4. Same argument applies, so find gcd(14,7)
5. Hence, gcd(91,287) = gcd(14,91) = gcd(7,14) = 7
```

Algorithm to Find gcd()

Lemma. Let $a = bq + r$, where a, b, q, and r are integers. Then $\gcd(a,b) = \gcd(b, r)$.

The Euclidean Algorithm

```
function gcd(a, b: positive integers)
  x <- a
  y <- b
  while (y != 0) {
    r <- x mod y
    x <- y
    y <- r
  } // end of loop to find gcd
  return x    //the last non-zero remainder
} // end of gcd function
```

Find: gcd(414, 662)

Integer Representations

- **Theorem**. Let b be a positive integer greater than 1. 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 non–negative integer, a_0, a_1, . . . , a_k are non–negative integers less than b, and $a_k \neq 0$

- The above representation of n is called the **base b expansion of n**, denoted by $(a_k a_{k-1} \ldots a_1 a_0)_b$

- Example I (octal): $(734)_8 = 7(8^2) + 3(8^1) + 4(8^0) = 476_{10}$

- Example II (binary): $1011001 = 2^6 + 2^4 + 2^3 + 2^0 = 89_{10}$

Hexadecimal — Base 16

- Hexadecimal or Base 16 digits are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A (10), B (11), C (12), D (13), E (14), and F (15)
- Given 4 bits, we can represent 16 different values, 0 – F:

0	-	0000	4	-	0100	8	-	1000	C	-	1100
1	-	0001	5	-	0101	9	-	1001	D	-	1101
2	-	0010	6	-	0110	A	-	1010	E	-	1110
3	-	0011	7	-	0111	B	-	1011	F	-	1111

- One byte is 8 bits, so a byte of information can be represented with two hexadecimal digits.
  For example: $0101\ 1101_2 \ = \ 5D_{16}$
- Example III:
  $$\begin{aligned}(2AE0B)_{16} &= 2(16^4) + 10(16^3) + 14(16^2) + 0(16^1) + 11(16^0)\\ &= 175,627_{10}\end{aligned}$$

Process to convert n_{10} to base b

1. Divide n by b to obtain a quotient and remainder:
   $$n = bq_0 + a_0, \quad 0 \le a_0 < b$$
   This remainder, a_0, is the rightmost digit in the base b expansion of n.

2. Divide q_0 by b: $q_0 = bq_1 + a_1, \quad 0 \le a_1 < b$
   This remainder, a_1, is the second digit from the right-hand side in the base b expansion of n.

3. Continue this process, successively dividing the quotients by b, obtaining additional base b digits as the remainders.

4. The process terminates when we obtain a quotient equal to zero

Conversion Algorithm

Constructing Base b Expansions

```
procedure base_b_expansion (n: positive integer){
  q <- n
  k <- 0

  while (q != 0) {
    a[k] <- q mod b
    q <- floor(q / b)
    k <- k + 1
  } // end conversion loop

  return a

} // end expansion
```

Conversion Practice

- Find the base 8 expansion of $(532)_{10}$
- Find the base 2 expansion of $(532)_{10}$
- Find the base 16 expansion of $(532)_{10}$

Arithmetic Operations — Addition

Addition in various bases is accomplished in a manner similar to base 10 addition

```
    binary              octal              hex

    101100               7340              29AC
 +  011010            +   521          +   A131
   -------              -----             -----
```

Representing Values in a Computer

Unsigned Integers

- non–negative integer representation
- used for such things as counting and memory addresses
- with k bits, exactly 2^k integers, ranging from 0 to $2^k - 1$ can be represented

Signed Integers

- If integers are stored in 8 bits, how many different bit patterns are there available to assign to various values?
- If we assign the bit pattern 0000 0000 to the value 0, how many are left for other values?
- There are different methods to deal with the "extra" bit pattern.

Signed Integer Representation

- Use the high–order (leftmost) bit to represent the **sign** of the number: 0 for positive, 1 for negative.

- All positive numbers (beginning with a 0 bit) are simply evaluated as is.

- If the first bit is 1 (signifying a negative number), there are several representation methods to consider.

Signed Integer Representation Schemes

1. **Signed Magnitude** — the other bits are evaluated to find the magnitude of the number (then make it negative).

2. **1's Complement** — flip (complement) the other bits before evaluating them to find the magnitude (then make it negative).

3. **2's Complement** — flip all the bits and add $00\ldots01$ before evaluating them to find the magnitude (then make it negative)

The following table is based upon a 4–bit representation. What happen when we add 1 and -1 in each representation?

bit pattern	Signed Magnitude	1's Complement	2's Complement
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000	−0	−7	−8
1001	**−1**	−6	−7
1010	−2	−5	−6
1011	−3	−4	−5
1100	−4	−3	−4
1101	−5	−2	−3
1110	−6	**−1**	−2
1111	−7	−0	**−1**

Section 3.7 — Applied Number Theory

- Theorem 1. (**linear combination**): If $a, b \in \mathbb{Z}^+$, then $\exists\ s, t \in \mathbb{Z} \ni$ gcd(a,b) = sa + tb
- s & t can be found by working backward through the divisions of the Euclidean Algorithm
- Express gcd(154,105) as linear combination of 252 and 198

```
Using the Euclidean Algorithm:
(2) 154 = 1(105) + 49
(1) 105 = 2(49) + 7
(0)  49 = 7(7) + 0        so gcd(154,105) = 7

Working Backwards:
by (1)  7 = 105 - 2(49)
by (2) 49 = 154 - 105
    so  7 = 105 - 2(154 - 105)
          = 3(105) - 2(154)
```

Linear Combination Example II

Find a linear combination of 252 and 198 which equals their gcd.

```
Using the Euclidean Algorithm:
(3) 252 = 1(198) + 54
(2) 198 = 3(54) + 36
(1)  54 = 1(36) + 18
(0)  36 = 2(18) + 0      so gcd(252,198) = 18

Working Backwards:
by (1)  18 = 54 - 1(36)
by (2)  36 = 198 - 3(54)
    so  18 = 54- 1(198 - 3(54))
           = 4(54) - 198
by (3)  54 = 252 - 1(198)
    so  18 = 4(252 - 198) - 198
           = 4(252) - 5(198)
```

Linear Combination Example III

Find a linear combination of 124 and 323 which equals their gcd.

```
Using the Euclidean Algorithm:
(7) 323 = 2(124) + 75
(6) 124 = 1(75) + 49
(5)  75 = 1(49) + 26
(4)  49 = 1(26) + 23
(3)  26 = 1(23) + 3
(2)  23 = 7(3) + 2
(1)   3 = 1(2) + 1
(0)   2 = 2(1) + 0

so gcd(124,323) = 1
```

You can make your life simpler by next rewriting the equations in terms of the remainders:

```
(7) 323 = 2(124) + 75      75 = 323 - 2(124)
(6) 124 = 1(75) + 49       49 = 124 - 1(75)
(5)  75 = 1(49) + 26       26 = 75 - 1(49)
(4)  49 = 1(26) + 23       23 = 49 - 1(26)
(3)  26 = 1(23) + 3         3 = 26 - 1(23)
(2)  23 = 7(3) + 2          2 = 23 - 7(3)
(1)   3 = 1(2) + 1          1 = 3 - 1(2)
(0)   2 = 2(1) + 0
```

Using Those Equations, We Obtain:

```
by (1)   1 = 3 - 1(2)


by (2)   2 = 23 - 7(3)
    so   1 = 3 - 1(23 - 7(3))
           = 8(3) - 23


by (3)   3 = 26 - 1(23)
    so   1 = 8(26 - 1(23)) - 23
           = 8(26) - 9(23)


by (4)  23 = 49 - 1(26)
    so   1 = 8(26) - 9(49 - 26)
           = 17(26) - 9(49)
```

```
by (5)  26 = 75 - 1(49)
    so   1 = 17(75 - 49) - 9(49)
           = 17(75) - 26(49)


by (6)  49 = 124 - 1(75)
    so   1 = 17(75) - 26(124 - 75)
           = 43(75) - 26(124)


by (7)  75 = 323 - 2(124)
    so   1 = 43(323 - 2(124)) - 26(124)
           = 43(323) - 112(124)
```

Linear Combination Example IV

Find a linear combination of 2002 and 2339 equal to their gcd.

Find gcd(2002, 2339):

Working Backwards. . .

Other Integer Results

- Lemma 1. If a, b, and $c \in \mathbb{Z}^+$ such that $\gcd(a,b) = 1$ and $a \mid bc$, then $a \mid c$.

- Lemma 2. If p is a prime and $p \mid a_1 a_2 \ldots a_n$ where each $a_i \in \mathbb{Z}$, then $p \mid a_i$ for some i.

- Theorem 2. Let $m \in \mathbb{Z}^+$ and let a, b, and $c \in \mathbb{Z}$. If $ac = bc \pmod m$ and $\gcd(c, m) = 1$, then $a \equiv b \pmod m$.

4.1 Mathematical Induction

- Similar to an infinite line of people, Person_1, Person_2, etc.

- A **secret** is told to Person_1, and each person **tells the secret** to the next person in line — if the former person hears it.

- Let P(n) be the proposition that Person_n **knows the secret**.

- Then P(1) is **true** since the secret is told to Person_1.

- P(2) is true since Person_1 **tells** Person_2, and so on.

- By the **Principle of Mathematical Induction**, every person in line learns the secret.

Mathematical Induction — Another Example

- Consider an infinite row of dominoes labeled 1, 2, 3, $\ldots$, n, where each domino is positioned to **knock the next one over** when it falls.

- Let $P(n)$ be the proposition that domino n is **knocked over**.

- If the first domino is **knocked over**, i.e., $P(1)$ is true, and if whenever the n^{th} domino is **knocked over**, it also **knocks over** the $(n+1)^{st}$ domino [i.e., $P(n) \to P(n+1)$ is true], then **all the dominoes are knocked over**.

Induction Proof — Big Picture

- Prove the proposition for the lower bound of n, say $n = 1$; i.e., show P(1) is true
- Assume for an **arbitrary** k that P(k) is true
- Set up a "proof machine" that demonstrates how to prove P(k+1) true when P(k) is true

---

You have then set up a way to "bootstrap" from P(1) as far as anyone would want to go.

We could simply keep applying the "proof machine" over and over, moving from P(1) to P(2) to P(3) to . . . well, as long as we wanted to!

Parts of an Induction Proof (**Required** in MAT 2345)

Label each section:

- **Basis or Base Case** (**BC**)
  Show the proposition is true for the lower bound of n
- **Inductive Hypothesis** (**IH**)
  Assume the proposition is true for an **arbitrary** k
- **Inductive Step** (**IS**)
  Show the proposition is true for (k+1), **using** the **inductive hypothesis**
  - give reasons for each step in the proof
  - usually begin with the LHS and show logical steps to reach the RHS

Prove using Induction:
Theorem. $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}, \quad \forall n \geq 0$

Prove using Induction:
Theorem. $\sum_{i=0}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}, \forall n \geq 0$

Prove using Induction:
Theorem. $\sum_{i=0}^{n} 2^i = 2^{n+1} - 1, \forall n \geq 0$

Prove using Induction:

Theorem. $\sum_{i=1}^{n} i(i+1)(i+2) = \frac{n(n+1)(n+2)(n+3)}{4}, \forall n \geq 1$

Prove using Induction:

Theorem. $5|(n^5 - n), \forall n \in \mathbb{N}$

Prove using Induction:
Theorem. $\sum_{i=0}^{n} i(i!) = (n+1)! - 1, \forall n \geq 1$

Prove using Induction:

Theorem. $6|(8^n - 2^n), \forall n \in \mathbb{N}$

Prove using Induction:

Theorem. $(x-y)|(x^n-y^n), \forall n, x, y \in \mathbb{N}$, and $y \geq 0$, $x \neq y$

Prove using Induction:

Theorem. $n! > 2^n$, $\forall\ n \geq 4$, $n \in \mathbb{N}$

The Pigeon–hole Principle

Theorem. If $n + 1$ balls $(n \geq 1)$ are put inside n boxes, then at least one box will contain more than one ball.

Example of Strong Induction

Theorem. All integers $n \geq 2$ can be written as a product of prime numbers (and 1)