§ 4.2 Integer Representations & Algorithms

Jessica Wei

Introduction

At the heart of encrypting a message is the idea of changing the representation of the characters so that if our message is intercepted, the string of characters is incomprehensible to the unintended reader. By now you are well aware that while humans communicate in words, computers communicate in numbers, and in order to understand how a computer can encrypt a string of characters, we need to understand how a computer represents integers. Some commonly used integer expansions are...

- Decimal Representation - base 10:
  $106 = 1 \cdot 100 + 0 \cdot 10 + 6 \cdot 1 = 1 \cdot 10^2 + 0 \cdot 10^1 + 6 \cdot 0 = (106)_{10}$
  $81 = 8 \cdot 10^1 + 1 \cdot 10^0 = (81)_{10}$
- Binary Representation - base 2:
  $(106)_{10} = 1\cdot 64+1\cdot 32+1\cdot 8+1\cdot 2 = 1\cdot 2^6+1\cdot 2^5+0\cdot 2^4+1\cdot 2^3+0\cdot 2^2+1\cdot 2^1+0\cdot 2^0 = (1101010)_2$
- Octal Representation - base 8
- Hexadecimal Representation - base 16

We will now delve deeper into these representations and device algorithms to convert between representations

Motivation: Cryptogrophy; to encrypt messages, we need to understand how computers represent numbers (data).

Integer Representations

THM 4.2.1 |
Let $b > 1 \in \mathbb{Z}$ and $n \in \mathbb{Z}^+$. Then $\exists!_{a_i} \in \mathbb{Z}$ satisfying $0 < a_i \leq b$ for $i = 0, 1, 2, ..., k, k \geq 0$ such that n has b-expansion

$$n = a_k b^k + a_{k-1} b^{k-1} + ... + a_1 b_1 + a_0 b_0$$

This is called the ab expansion of n, and a_i is called the coefficient of b^i

Example 1. What is the binary representation of the following numbers?

(a) 3
$= 1 \cdot 2^1 + 1 \cdot 2^0$
Answer: $(11)_2$

(b) 17
$1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$
Answer: $(10001)_2$

Example 2. What is the decimal expansion of each of the following?

(a) $(101011111)_2$
$= 1 \cdot 2^8 + 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$
$= 256 + 64 + 16 + 8 + 4 + 2 + 1$
Answer: 351_{10}

(b) $(7016)_8$
$= 7 \cdot 8^3 + 0 \cdot 8^2 + 1 \cdot 8^1 + 6 \cdot 8^0$
Answer: 3598_{10}

(c) $(2AE0B)_{16}$
$= 2 \cdot 16^4 + 10 \cdot 16^3 + 14 \cdot 16^2 + 0 \cdot 16^1 + 11 \cdot 16^0$
Answer: $175,627_{10}$

ALGORITHM | Converting Base -10 to Base-2,-8-,16
Suppose $n, b \in \mathbb{Z}^+$ where n is in decimal expansion and b is the base you wish to convert to. Then...

1. Divide n by b to obtain $n = b \cdot q + a_0$.
2. Keep a_0 as the rightmost digit in the base b-expansion.
3. Repeat 1. - 2. with the result a_i as the next digit in the b-expansion until q = 0.

Example 3.

(a) Find the octal representation of $(12345)_{10}$
$12,345 = 1,543 \cdot 8 + 1$
$1,543 = 24 \cdot 8 + 0$
$24 = 3 \cdot 8 + 0$
$3 = 0 \cdot +3$
Answer: 3001_8

PSEUDOCODE | Constructing Base b expansions
INPUT: $n > 0, b > 10$integers
OUTPUT: $(a_{k-1}, a_{k-2}, ..., a_1, a_0)-$ base b expansion

```
bExpansion(n,b):
    q = n
    r = []
    while q != 0:
        r[k] = a mod b #keep remainder
        q = q div b #computing new quotient for next loop
        k++
    return (a_k-1, a_k-2, ..., a_1, a_0)
    #reverse order of list r
```

ALGORITHM | Converting from Base-8 or Base-16 to Base-2
Suppose $n \in \mathbb{Z}^+$ with octal or hexadecimal expansion $(a_{k-1}, ak-2, ..., a_1, a_0)$.

1. If $b = 8$: Convert each digit a_i to a binary block of three digits.
   If $b = 16$: Convert each digit a_i to a binary block of four digits.
2. Concatenate the blocks.

Example 4. Find the binary expansion of each representation.

(a) $(765)_8$
$5 = 101$
$6 = 110$
$7 = 111$
Answer: 111110101_2

(b) $(A8D)_{16}$
$D = 1101$
$8 = 1000$
$A = 1010$
Answer: 101010001101_2

Integer Operations

ALGORITHM | Addition of Binary Numbers
Motivation: $9 + 5 = 13$ where

$$9 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \Rightarrow 1001$$

$$5 = 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \Rightarrow 0101$$

$$1001 + 0101 \Rightarrow 1110 = 13$$

Suppose $a = (a_{k-1}, a_{k-2}, ..., a_1, a_0)_2$ and $u = (u_{k-1}, u_{k-2}, ..., u_1, u_0)_2$ are the binary representations of decimal numbers a and u.

1. For $i = 0, 1, ..., k-1$ perform $a_i + u_i \mod 2$ plus any carry from the $a_{i-1} + u_{i+1}$.

PSEUDOCODE | Adding two numbers with base-2 expansions
INPUT: $a = (a_{k-1}, a_{k-2}, ..., a_1, a_0)$, $u = (u_{k-1}, u_{k-2}, ..., u_1, u_0)$
OUTPUT: $(s_{k-1}, s_{k-2}, ..., s_1, s_0)_2$ sum of $a + u$

```
binaryAdd(a,u):
    c = 0 #carry
    for i in 0:k-1:
        s_i = (a_i + u_i + c) mod 2
        c = (a_i + u_i + c) div 2
    s_k = c
return (s_k, s_k-1, s_k-2, ..., s_1, s_0)
```

RUN-THROUGH:

ALGORITHM | Multiplication of Binary Numbers
Motivation: $11 \cdot 3 = 33$ where

$$11 = 1011$$

$$3 = 0011$$

$$\Rightarrow 1011 + 10110 + 000000 + 0000000 = 0100001 = 33$$

PSEUDOCODE | Multiplying two numbers with base-2 expansions
INPUT: $a = (a_{k-1}, a_{k-2}, ..., a_1, a_0)$, $u = (u_{k-1}, u_{k-2}, ..., u_1, u_0)_2$
OUTPUT: $(s_{k-1}, s_{k-2}, ..., s_1, s_2)_2$ product of $a \cdot u$

```
binaryProduct(a,u):
    c = [] #container for binary coefficients of each product
    s = 0
        for i = 0:k-1:
            if b_j = 1:
                x = a + j zeros appended to the end
                c.append(x)
            else:
                c.append(0)
        for i = 0:k-1:
            s = binaryAdd(s,c[i])
    return s
```

ALGORITHM | Modular Exponentiation
GOAL: $b^n \mod m$ where b, m, and n are large.

1. Find the binary expansion of $n, n = a_{k-1}2^{k-1} + ... + a_1 2 + a_0$.
2. Use Corollary to THM 4.1.4 to compute $b^n \mod m$.

$$\begin{gathered}
b^n \mod m \\
= b^{a_{k-1}2^{k-1} + ... + a_1 2 + a_0} \mod m \\
= b^{a_{k-1}2^{k-1}} \cdot b^{a_{k-2}2^{k-2}} ... b^{a_1 2} \cdot b^{a_0} \mod m \\
= [(b^{a_{k-1}2^{k-1}} \mod m)(b^{a_{k-2}2^{k-2}} ... b^{a_1 2} \cdot b^{a_0} \mod m)] \mod m \\
= [(b^{a_{k-1}2^{k-1}} \mod m)((b^{a_{k-2}2^{k-2}} \mod m) b^{a_{k-3}2^{k-3}} ... b^{a_1 2} \cdot b^{a_0} \mod m) \mod m \\
...
\end{gathered}$$

Example 5. Compute $3^{11} \mod 2$.
Binary expansion of $11 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$
$\therefore 3^{11} \mod 2 = 3^{8+2+1} \mod 2 = 3^8 \cdot 3^2 \cdot 3^1 \mod 2$
$= (3^8 \mod 2)(3^2 \cdot 3^1 \mod 2) \mod 2$
$= (3^8 \mod 2)[(3^2 \mod 2)(3^1 \mod 2) \mod 2] \mod 2$
$= (3^8 \mod 2)[1 \cdot 1 \mod 2] \mod 2$
$= (6561 \mod 2)(1) \mod 2 = 1 \cdot 1 \mod 2$
Answer: 1

PSEUDOCODE | Modular Exponentiation
INPUT: b-integer, m-integer, $n = (a_{k-1}, a_{k-2}, ..., a_1, a_0)_2$
OUTPUT: $b^n \mod m$

```
modExp(b,n,m):
    x = 1
    p = b mod m
        for i = 0:k-1:
            if a_i = 1:
                x = x * p mod m
            p = p * p mod m
        return x
```