CSE 20
DISCRETE MATH

Winter 2017

http://cseweb.ucsd.edu/classes/wi17/cse20-ab/
Today's learning goals

• Describe and use algorithms for integer operations based on their expansions
• Define and use the DIV and MOD operators.
• Relate algorithms for integer operations to bitwise boolean operations
• Correctly use XOR and bit shifts
To change your remote frequency
1. Press and hold power button until flashing
2. Enter two-letter code
3. Checkmark / green light indicates success

How many people were in your group for HW1?

A. I worked alone.
B. 2
C. 3
D. I joined this class late and didn’t submit HW1.
Base expansion

Notation: for positive integer $n$

Write

$$\left( a_{k-1} \ldots a_1 a_0 \right)_b$$

when

$$n = a_{k-1}b^{k-1} + \ldots + a_1b + a_0$$

Base $b$ expansion of $n$
**Algorithm:** constructing base b expansion  
*Rosen p. 249*

**Input** n,b  
**Output** k, coefficients in expansion

- English description.
  
  Find k by computing successive powers of b until find smallest k such that
  
  $$b^{k-1} \leq n < b^k$$
  
  For each value of i from 1 to k
  
  Set $a_{k-i}$ to be the largest number between 0 and b-1 for which $a_{k-i} b^{k-i} \leq n$.
  
  Update current value remaining $n := n - a_{k-i} b^{k-i}$
Algorithm: constructing base b expansion

Input n,b  
Output k, coefficients in expansion

• English description.
  Idea: Find smallest digit first, then next smallest, etc.
  .... but how?

• Pseudocode.
Bases and Divisibility

Remember that \((17)_{10} = (122)_{3}\)

What's the lowest-order symbol (in the 1's place) in the base 3 expansion of 18?

A. 0  
B. 1  
C. 2  
D. 3  
E. I don't know.
Bases and Divisibility

Useful fact:
if n is a multiple of b, the base b expansion of n ends in 0.

Why?
Theorem: For \( n \) an integer and \( d \) a positive integer, there are unique integers \( q \) and \( r \) with \( 0 \leq r < d \) and \( n = qd + r \).

Notation: \( q = n \, \text{div} \, d \) \quad \text{Remainder} \quad r = n \, \text{mod} \, d \)

What is \( 24 \, \text{div} \, 5 \)? What is \( 15 \, \text{mod} \, 5 \)?

A. 4, 0 \quad \text{B.} \ 3, 0 \quad \text{C.} \ 4, 3 \quad \text{D.} \ 3, 3 \quad \text{E.} \ \text{I don't know.}
When $k > 0$

$$n = a_k b^k + a_{k-1} b^{k-1} + \ldots + a_1 b + a_0$$

$$= b \left( a_k b^{k-1} + a_{k-1} b^{k-2} + \ldots + a_1 \right) + a_0$$

$q = n \div d$

$r = n \mod d$
Bases and Divisibility

Useful fact:
if \( n = (a_{k-1}...a_0)_b \) then \( bn = (a_{k-1}...a_00)_b \)

Why?
Algorithm: constructing base b expansion  

Input $n, b$  
Output $k$, coefficients in expansion  

• English description. 

Compute $n \mod b$ to obtain $a_0$.  
Update value $n := n \div b$ of integer whose expansion we need.  
Repeat.
**Algorithm:** constructing base b expansion  
*Rosen p. 249*

**Input**  \( n, b \) \quad **Output**  \( k, \text{coefficients in expansion} \)

**Pseudocode:**

procedure \textit{base b expansion}(n, b : pos ints with \( b > 1 \))

1. \( q := n \)
2. \( k := 0 \)
3. while \( q \neq 0 \)
4. \( a_k := q \mod b \)
5. \( q := q \div b \)
6. \( k := k + 1 \)
7. return \((a_{k-1}, \ldots, a_1, a_0)\)
Algorithm: constructing base b expansion \( \text{Rosen p. 249} \)

\[
\text{procedure } \text{base } b \text{ expansion}(n, b : \text{ pos ints with } b > 1) \\
\]

1. \( q := n \)
2. \( k := 0 \)
3. while \( q \neq 0 \)
4. \( a_k := q \mod b \)
5. \( q := q \div b \)
6. \( k := k + 1 \)
7. return \((a_{k-1}, \ldots, a_1, a_0)\)

<table>
<thead>
<tr>
<th>( n )</th>
<th>( b )</th>
<th>( q )</th>
<th>( k )</th>
<th>( a_k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>3</td>
<td>17</td>
<td>0</td>
<td>( 17 \mod 3 = 2 )</td>
</tr>
<tr>
<td>( 17 \div 3 = 5 )</td>
<td>1</td>
<td>( 5 \mod 3 = 2 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 5 \div 3 = 1 )</td>
<td>2</td>
<td>( 1 \mod 3 = 1 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 1 \div 3 = 0 )</td>
<td>3</td>
<td>return!</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\text{Definite? Finite? Correct?}
Properties of binary expansions

- n is odd *exactly when* coefficient of $2^0$ in expansion is 1
- The **biggest** integer value whose binary representation has 4 bits is …

A. 4  
B. 8  
C. 16  
D. 1111  
E. None of the above.
Properties of binary expansions

• n is odd exactly when coefficient of $2^0$ in expansion is 1
• The smallest integer value whose binary representation has 4 bits is ...

A. 0  
B. 4  
C. 8  
D. 16  
E. None of the above.
Fixed width "binary expansions"
with 4 bits

(0000)_2 = 0
(0001)_2 = 1
(0010)_2 = 2
...
(1110)_2 = 14
(1111)_2 = 15
Representing more?

• Base b expansions can express any positive integers.

• What about
  • negative integers?
  • rational numbers?
  • other real numbers? stay tuned for CSE 30, CSE 140
Back to arithmetic

In base $b$,

\[
\begin{array}{c}
s_{k-1} \ldots s_1 s_0 \\
+ t_{k-1} \ldots t_1 t_0
\end{array}
\]

Basic operations: one symbol addition, carry

Rosen p. 251, 826
# Arithmetic + Representations

For decimal

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>9</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>
Arithmetic + Representations

For binary

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Alternatively,

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Arithmetic + Representations

Alternatively,

Half adder logic circuit

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Computer bit operations

Rosen p. 11

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>x OR y</th>
<th>x AND y</th>
<th>x XOR y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

"x OR y is 1 if at least one of x or y is 1"

"x AND y is 1 if both x and y are 1"

"x XOR y is 1 if exactly one of x and y is 1"
What is the bitwise AND of the strings 0011 and 0101?
A. 0001    D. 0100
B. 0111    E. None of the above
C. 0010
What is the bitwise XOR of the strings 0011 and 0101?
A. 0110
B. 0000
C. 0111
D. 0100
E. None of the above
Logic

- Use gates and circuits to express arithmetic.

- Precisely express theorems and invariant statements.

- Make valid arguments to prove theorems.
Definitions

- **Proposition**: declarative sentence that is T or F (not both)
- **Propositional variable**: variables that represent propositions.
- **Compound proposition**: new propositions formed from existing propositions using logical operators.
- **Truth table**: table with 1 row for each of the possible combinations of truth values of the input and an additional column that shows the truth value of the result of the operation corresponding to a particular row.

*Rosen p. 2-4*
Definitions

- **Proposition**: declarative sentence that is T or F (not both).
- **Propositional variable**: variables that represent propositions.
- **Compound proposition**: new propositions formed from existing propositions using logical operators.
- **Truth table**: table with 1 row for each of the possible combinations of truth values of the input and an additional column that shows the truth value of the result of the operation corresponding to a particular row.

Is the sentence "This sentence is false" true or false?
How many rows are in the truth table for \((p \lor q) \lor (p \lor r)\)?

A. 1  
B. 2  
C. 3  
D. 4  
E. None of the above
Truth tables

- Can use truth table to compute value of compound variable.

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>r</th>
<th>p v q</th>
<th>p v r</th>
<th>(p v q) v (p v r)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Truth tables

• Can use truth table to compute value of compound variable.
• Also, can specify logical operator by truth table.

• Next time: how to prove two tables are equivalent?
Reminders

- Homework 2 due Sunday at noon
  - Bit operations, circuits, logic, propositions.

- Office hours