CSE 20
DISCRETE MATH

Winter 2017

HW 1 solutions on Piazza
HW1 scores + comments by tomorrow

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

About you

MANDE B-210: AC PETER 108: AC

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

Rosen p. 246

Notation: for positive integer n

Write

$$(a_{k-1} \dots a_1 \, a_0)_b$$

when

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

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

  $$\mathbf{b^{k-1} \leq n < b^{k}}$$

  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 *Rosen p. 249*

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

Rosen p. 237-239

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

Rosen p. 237-239

Useful fact:

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

Why?

Reminder: Divisibility

Rosen p. 237-239

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 **div** d   r = n **mod** d

Quotient when divide n by b

Remainder when divide n by b

What is 24 **div** 5? What is 15 **mod** 5?

A. 4, 0
B. 3, 0
C. 4, 3
D. 3, 3
E. I don't know.

Bases and Divisibility

Rosen p. 237-239

When k>0

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

$$= b\,(a_k b^{k-1} + a_{k-1} b^{k-2} + \dots + a_1) + a_0$$

d

q = n div d

r = n mod d

Bases and Divisibility

Rosen p. 237-239

Useful fact:

if $n = (a_{k-1}\ldots a_0)_b$ then $bn = (a_{k-1}\ldots a_0 0)_b$

Why?

Algorithm: constructing base b expansion *Rosen p. 249*

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 **Output** k, coefficients in expansion

Pseudocode:

procedure *base b expansion*$(n, b:$ pos ints with $b > 1)$

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

Algorithm: constructing base b expansion *Rosen p. 249*

procedure *base b expansion*$(n, b:$ pos ints with $b > 1)$

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

n	*b*	*q*	*k*	a_k
17	**3**	17	0	17 **mod** 3 = 2
		17 **div** 3 = 5	1	5 **mod** 3 = 2
		5 **div** 3 = 1	2	1 **mod** 3 = 1
		1 **div** 3 = 0	3	return!

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

Rosen p. 251, 826

In base b,

$$
\begin{array}{r}
s_{k-1} \ \dots \ s_1 \ s_0 \\
+\ t_{k-1} \ \dots \ t_1 \ t_0 \\
\hline
\end{array}
$$

Basic operations: one symbol addition, carry

Arithmetic + Representations

Rosen p. 251,826

For decimal

	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9	0
2	2	3	4	5	6	7	8	9	0	1
3	3	4	5	6	7	8	9	0	1	2
4	4	5	6	7	8	9	0	1	2	3
5	5	6	7	8	9	0	1	2	3	4
6	6	7	8	9	0	1	2	3	4	5
7	7	8	9	0	1	2	3	4	5	6
8	8	9	0	1	2	3	4	5	6	7
9	9	0	1	2	3	4	5	6	7	8

Arithmetic + Representations

Rosen p. 251,826

For binary

	0	1
0	0	1
1	1	0

Arithmetic + Representations

Rosen p. 251,826

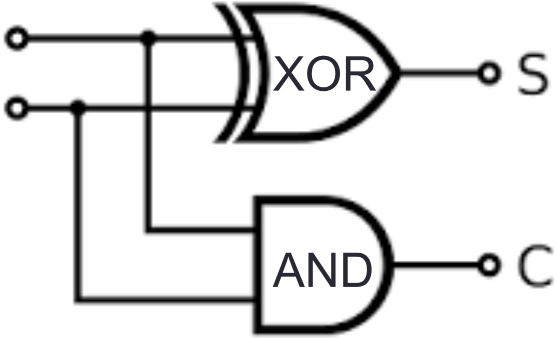
Alternatively,

Input		Output	
x	**y**	**Sum**	**Carry**
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Arithmetic + Representations

Rosen p. 251,826

Alternatively,



Input		Output	
x	**y**	**Sum**	**Carry**
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Half adder logic circuit

Computer bit operations *Rosen p. 11*

x	y	$x \vee y$ x OR y	$x \wedge y$ x AND y	$x \oplus y$ x XOR y
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

"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"

Computer bit operations

Rosen p. 11

x	y	x ∨ y x OR y	x ∧ y x AND y	x ⊕ y x XOR y
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

What is the **bitwise** AND of the strings 0011 and 0101?

A. 0001
B. 0111
C. 0010
D. 0100
E. None of the above

Computer bit operations

Rosen p. 11

x	y	x ∨ y x OR y	x ∧ y x AND y	x ⊕ y x XOR y
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

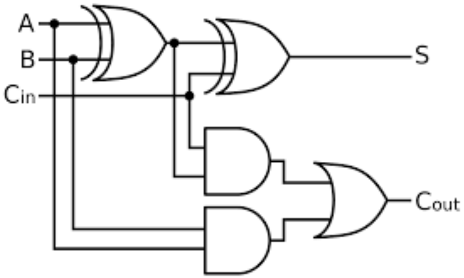
What is the **bitwise** XOR of the strings 0011 and 0101?

A. 0110
B. 0000
C. 0111
D. 0100
E. None of the above

Logic

Rosen Section 1.1

- Use gates and circuits to express arithmetic.



- Precisely express theorems and invariant statements.
- Make valid arguments to prove theorems.

Definitions

Rosen p. 2-4

- **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.

Definitions

Is the sentence "This sentence is false" true or false?

- **Proposition**: declarative senten… n)
- **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.

Truth tables

How many rows are in the truth table for **(p v q) v (p v 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.

p	q	r	p v q	p v r	(p v q) v (p v r)
1	1	1			
1	1	0			
1	0	1			
1	0	0			
0	1	1			
0	1	0			
0	0	1			
0	0	0			

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