CSE 20
DISCRETE MATH

Fall 2020

http://cseweb.ucsd.edu/classes/fa20/cse20-a/

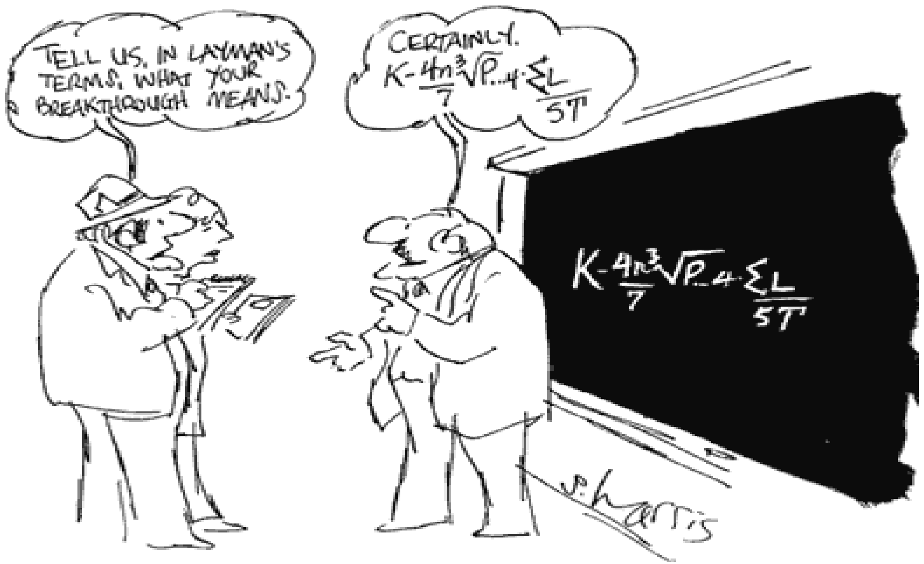
Today's learning goals

- Trace an algorithm specified in pseudocode
- Define the base expansion of a positive integer, specifically decimal, binary, hexadecimal, and octal.
- Convert between expansions in different bases of a positive integer.
- Define and use the **div** and **mod** operators.

Learning goals

In the past two classes, when have we used numbers?

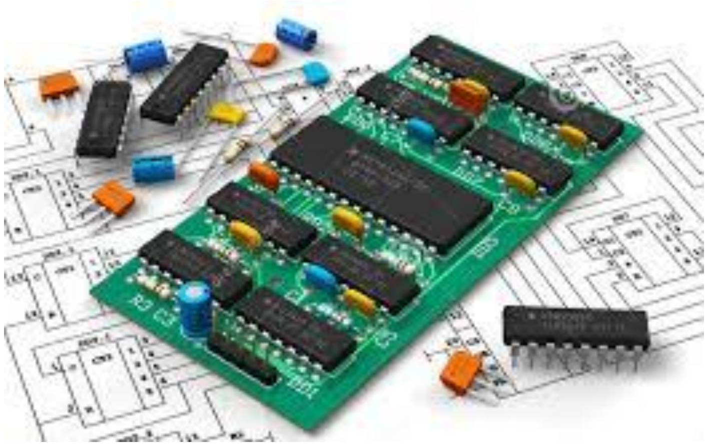
Multiple Representations



Integer representations

Different contexts call for different representations.

Base 10



Base 2

Base b expansion of n

Rosen p. 246

Also known as **positional representation** of positive integers

Definition (Rosen p. 246) For b an integer greater than 1 and n a positive integer, the **base b expansion of n** is

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

where k is a positive integer, $a_0, a_1, \ldots, a_{k-1}$ are nonnegative integers less than b, $a_{k-1} \neq 0$, and

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

Using the terminology from last class: the base b expansion of n is a string over the alphabet $\{x \in \mathbb{N} \mid x < b\}$ and whose leftmost character is nonzero.

Base b expansion

In what base **could** this expansion be

$$(1401)_?$$

A. Binary (base 2)
B. Octal (base 8)
C. Decimal (base 10)
D. Hexadecimal (base 16)
E. More than one of the above

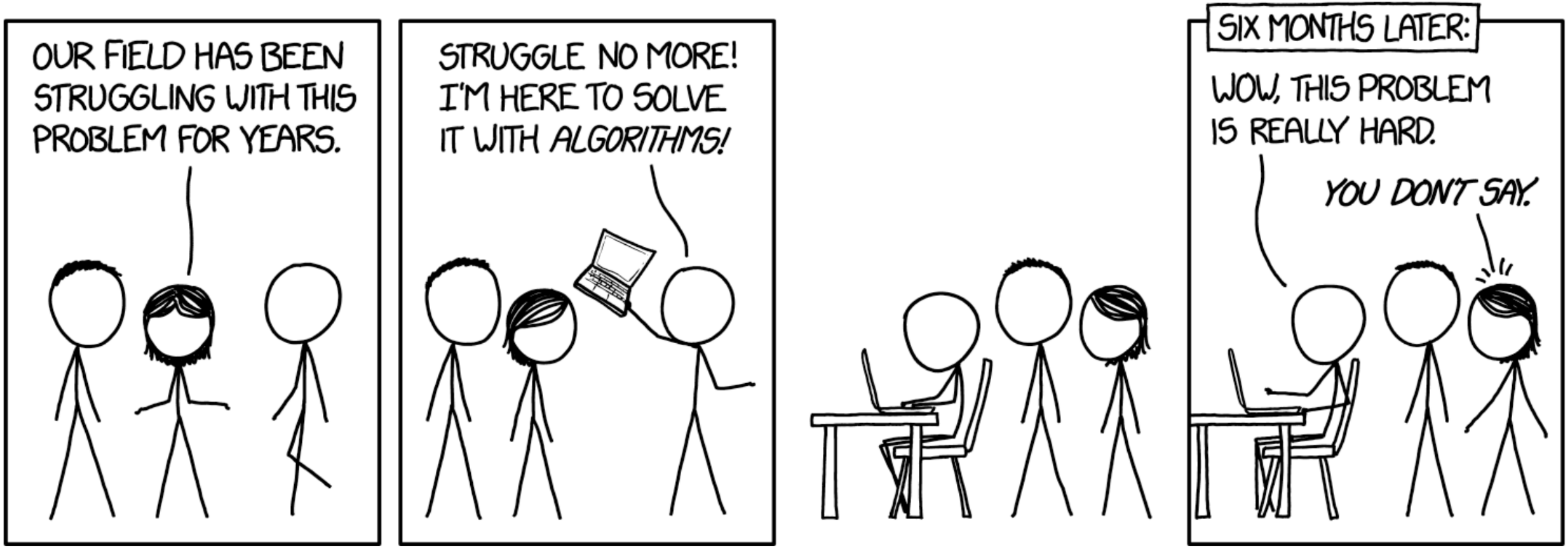
Base b expansion

In what base **could** this expansion be

$$(1401)_?$$

A. Binary (base 2)
B. Octal (base 8) $(1401)_8 = 1*8^3+4*8^2+1=(769)_{10}$
C. Decimal (base 10) $(1401)_{10} =1*10^3+4*10^2+1=1401$
D. Hexadecimal (base 16) $(1401)_{16} =1*16^3+4*16^2+1=5121$
E. More than one of the above

Converting between bases

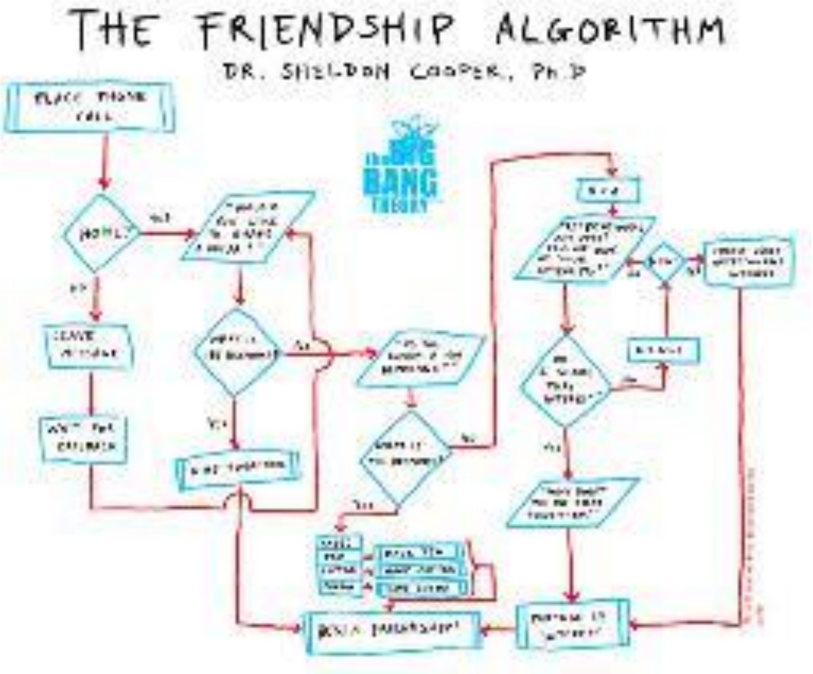


xkcd

Algorithm?

Rosen 3.1 p. 191

Finite sequence of precise instructions for solving problem.



Algorithm: Pseudocode

Appendix

Finite sequence of precise instructions for solving problem.

```
procedure log(n: a positive integer)
r := 0
while n > 1
   r := r + 1
   n := n div 2
return r {r holds the result of the log operation}
```

At the end of running *log(6)* what values are in the variables *r* and *n*?

A. r = 6, n = 0
B. r = 6, n = 6
C. r = 2, n = 0
D. r = 2, n = 1
E. None of the above.

Algorithm: constructing base b expansion

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

- English description.

- Pseudocode.

Algorithm 1: constructing base b expansion

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

- English description.

  Initialize value remaining to be n

  Find biggest power of b that is less than or equal to value remaining.

  Increment appropriate coefficient.

  Update value remaining by subtract this power of b from it.

  Repeat until value remaining is 0.

Ternary representation of 17

A. $(17)_3$
B. $(211)_3$
C. $(122)_3$
D. $(221)_3$
E. $(112)_3$

Algorithm 1: constructing base b expansion

Calculating base b expansion, from left

procedure *baseb1*$(n, b\colon$ positive integers with $b > 1)$
$v := n$
$k := logb(n, b) + 1$
for $i := 1$ **to** k
  $a_{k-i} := 0$
  while $v \geq b^{k-i}$
    $a_{k-i} := a_{k-i} + 1$
    $v := v - b^{k-i}$
return $(a_{k-1}, \ldots, a_0)\{(a_{k-1} \ldots a_0)_b$ is the base b expansion of $n\}$

a_{k-1} is coefficient of biggest power of b that is less than n
Thus: k is 1 more than integer part of $\log_b n$

Algorithm 2: constructing base b expansion

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

Idea: Find smallest digit first, then next smallest, etc.

…. **but how?**

Rosen p. 249

Bases and 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 $n = dq + r$. **Notation:** $q = n$ **div** d and $r = n$ **mod** d

When k>1

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

$$n = \underbrace{b}_{d}\,\underbrace{(a_{k-1}b^{k-2} + \ \dots + a_1)}_{q\,=\,n \text{ div } d} + \underbrace{a_0}_{r\,=\,n \text{ mod } d}$$

d

q = n div d

r = n mod d

Algorithm 2: constructing base b expansion

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

Idea: Use n **mod** b to compute least significant digit.

Use n **div** b to compute new integer whose expansion we need. Repeat.

Algorithm 2: constructing base b expansion

Calculating base b expansion, from right

```
procedure baseb2(n, b: positive integers with b > 1)
q := n
k := 0
while q ≠ 0
    a_k := q mod b
    q := q div b
    k := k + 1
return (a_{k-1}, ..., a_0){(a_{k-1}, ..., a_0)_b is the base b expansion of n}
```

n	b	q	k	a_k	$q \neq 0$?

Representing more

- Base b expansions can express any **positive integers**

- What about
  - Zero?
  - negative integers?
  - rational numbers?
  - other real numbers?

For next time

- Read website carefully

http://cseweb.ucsd.edu/classes/fa20/cse20-a/

- No pre-class reading for next lecture

There are 10 types of people in the world: those who understand ternary, those who don't, and those who mistake it for binary