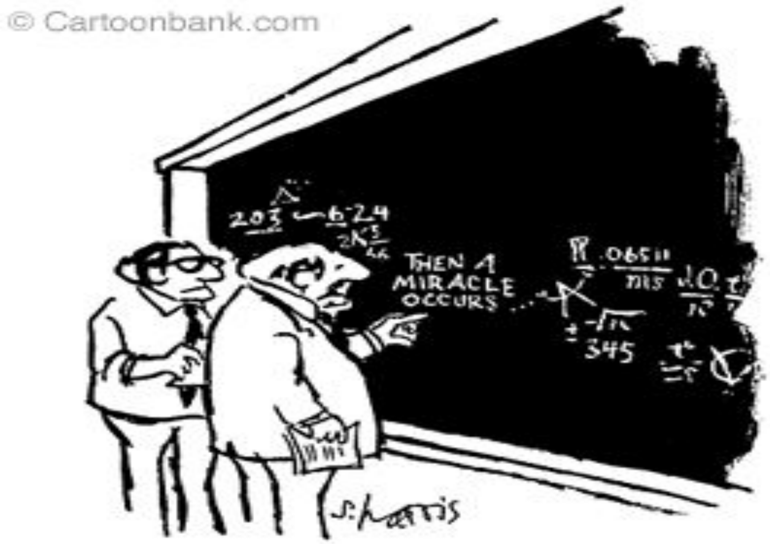
CSE 20
DISCRETE MATH

Winter 2021

http://cseweb.ucsd.edu/classes/wi21/cse20-ab/

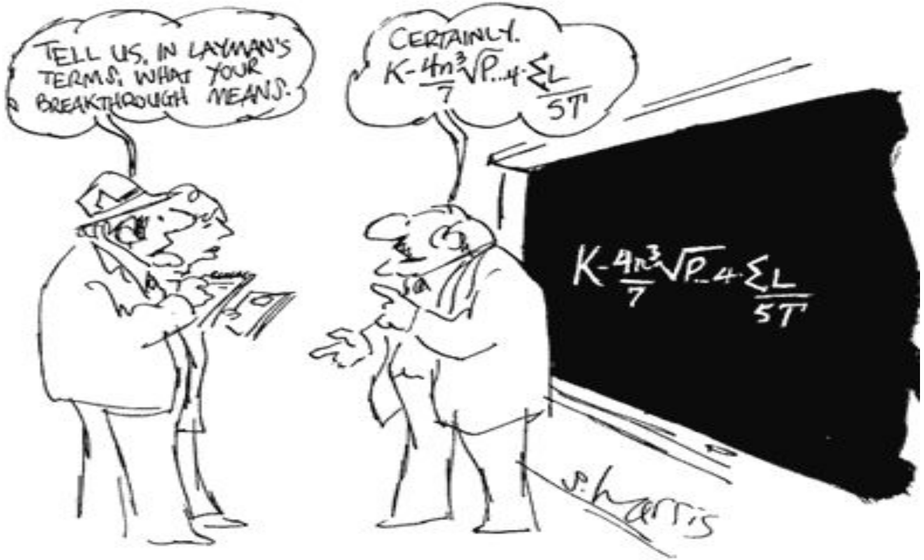
Learning goals

Technical Skepticism



"I think you should be more explicit here in step two."

Multiple Representations



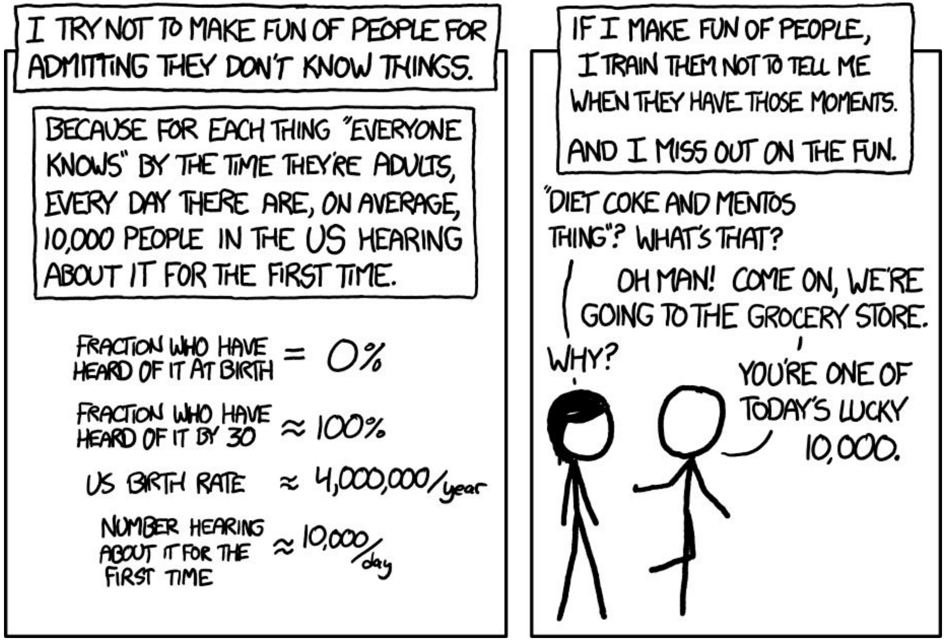
About the team

Prof Mia Minnes – “Minnes” *rhymes with* Guinness
(cseweb.ucsd.edu/~minnes, minnes@eng.ucsd.edu)

4 TAs + 10 tutors

** Office hours available: drop-in during times listed on calendar on class website. Ask questions about class examples, assignment questions, or other CS topics. **
http://cseweb.ucsd.edu/classes/wi21/cse20-ab/

Introductions



xkcd.com

Education research

Educational Research

This class is participating in research to understand an array of specific classroom and learning experience that students have in response to the pedagogical and curricular decisions instructors make and to address the following research questions:

- What pedagogies lead to better learning outcomes, and for which students?
- What educational practices increase the persistence and success of students, particularly those from underrepresented groups?
- What student practices lead to increased learning and success in real-world settings?

Answers to these questions will inform teaching practice at UC San Diego, and also have the potential to contribute to the global knowledge base of how to improve student learning in a large university setting.

Specifically for this quarter, CSE 20 is participating in a project on academic integrity and in a project analyzing different types of assignments.

Details about the academic integrity project will be sent to your @ucsd.edu email from integrity@ucsd.edu .

Details on the project studying different types of assignments are in this document. In particular, if you consent to participate in this study, no action is needed. If you DO NOT consent to participate in this study, or you choose to opt-out at any time during the quarter, please submit this form online. Your instructor will not have access to the list of students who opted out until after grades are posted. Note that you must separately opt-out of the study for each course involved in this study.

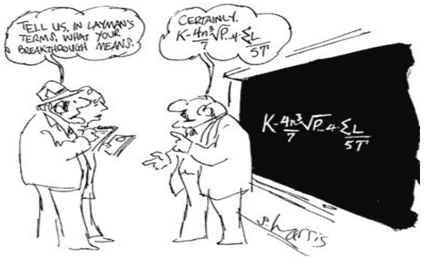
Monday's learning goals

- Practice with some **definitions** and **notation**
- Explore mathematical **definitions** related to a specific **application** (Netflix)

n-tuples, preferences, and Netflix

NETFLIX

Multiple Representations



What data should we encode about each Netflix account holder to help us make effective recommendations?

n-tuples, preferences, and Netflix

n-tuple	(x_1, x_2, x_3)	The 3-tuple of x_1, x_2, and x_3
	$(3, 4)$	The 2-tuple or ordered pair of 3 and 4

Person	Fyre	Frozen II	Picard
P_1	✗	•	✓
P_2	✓	✓	✗
P_3	✓	✓	✓
P_4	•	✗	✓

✗ Did not like
• No preference
✓ Liked

n-tuples, preferences, and Netflix

n-tuple	(x_1, x_2, x_3)	The 3-tuple of x_1, x_2, and x_3
	$(3, 4)$	The 2-tuple or ordered pair of 3 and 4

Person	Fyre	Frozen II	Picard	Ratings written as a 3-tuple
P_1	✗	•	✓	$(-1, 0, 1)$
P_2	✓	✓	✗	$(1, 1, -1)$
P_3	✓	✓	✓	$(1, 1, 1)$
P_4	•	✗	✓	

✗ Did not like: represent with -1
• No preference: represent with 0
✓ Liked: represent with 1

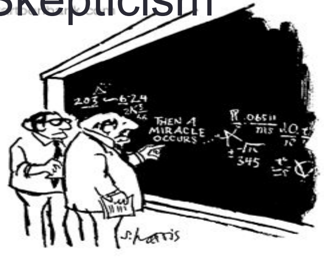
How similar are people's preferences?

Which of P_1, P_2, P_3 has movie preferences most similar to P_4?

A: P1
B: P2
C: P3
D: There is a tie

Person	Fyre	Frozen II	Picard
P_1	✗	•	✓
P_2	✓	✓	✗
P_3	✓	✓	✓
P_4	•	✗	✓

Technical Skepticism



"I think you should be more explicit here in step two."

Peterson 110: AD
To change your remote frequency

1. Press and hold power button until flashing
2. Enter two-letter code
3. Checkmark / green light indicates success

One approach: functions

function definition	$f(x) = x + 4$	Define f of x to be $x + 4$
function application	$f(7)$	f of 7 **or** f applied to 7 **or** the image of 7 under f
	$f(z)$	f of z **or** f applied to z **or** the image of z under f
	$f(g(z))$	f of g of z **or** f applied to the result of g applied to z

Page 2 of worksheet:

This page has some useful notation that will be used throughout the course. Find the definitions for each of these terms by looking in the index of the course textbook.

Define the following functions whose inputs are ordered pairs of 3-tuples each of whose components comes from the set $\{-1, 0, 1\}$

$$d_1(\ (x_1, x_2, x_3), (y_1, y_2, y_3)\ ) = \sum_{i=1}^{3} ((|\ x_i - y_i\ | + 1) \textbf{ div } 2)$$

Person	Fyre	Frozen II	Picard	Ratings written as a 3-tuple
P_1	✗	•	✓	$(-1, 0, 1)$
P_2	✓	✓	✗	$(1, 1, -1)$
P_3	✓	✓	✓	$(1, 1, 1)$
P_4	•	✗	✓	

$$d_1(P_4, P_1)$$

roster method	$\{43, 7, 9\}$	The set whose elements are 43, 7, and 9

Define the following functions whose inputs are ordered pairs of 3-tuples each of whose components comes from the set $\{-1, 0, 1\}$

Person	Fyre	Frozen II	Picard	Ratings written as a 3-tuple
P_1	✗	•	✓	$(-1, 0, 1)$
P_2	✓	✓	✗	$(1, 1, -1)$
P_3	✓	✓	✓	$(1, 1, 1)$
P_4	•	✗	✓	

$$d_2(\ (x_1, x_2, x_3), (y_1, y_2, y_3)\ ) = \sqrt{\sum_{i=1}^{3}(x_i - y_i)^2}$$

$$d_2(P_4, P_1)$$

Define the following functions whose inputs are ordered pairs of 3-tuples each of whose components comes from the set $\{-1,0,1\}$	
$d_1(\ (x_1,x_2,x_3),(y_1,y_2,y_3)\ ) = \sum_{i=1}^{3} ((\mid x_i - y_i \mid + 1) \textbf{ div } 2)$	$d_2(\ (x_1,x_2,x_3),(y_1,y_2,y_3)\ ) = \sqrt{\sum_{i=1}^{3} (x_i - y_i)^2}$

$d_1(P_4, P_1)$	$d_1(P_4, P_2)$	$d_1(P_4, P_3)$
$d_2(P_4, P_1)$	$d_2(P_4, P_2)$	$d_2(P_4, P_3)$

Wednesday's learning goals

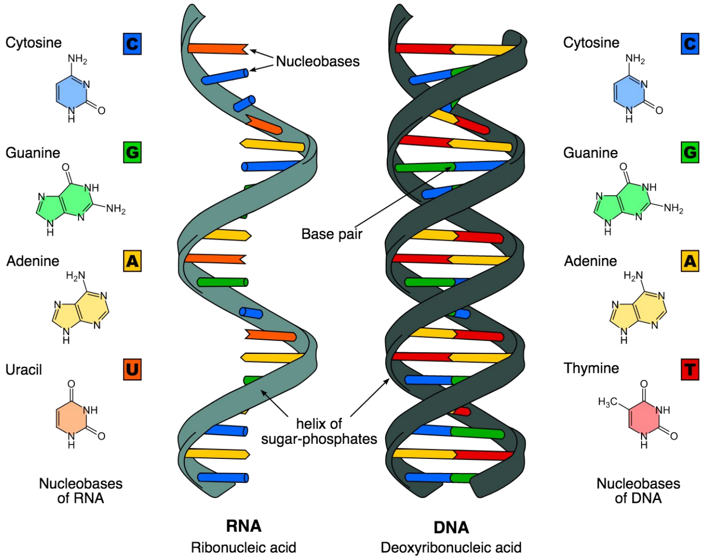
- Define data types: set, n-tuple, string (over specific alphabet)
- Define sets and functions in multiple ways

Types

- set: unordered, repetition doesn't matter
- n-tuple: ordered, repetition matters, fixed length
- string: ordered, repetition matters, arbitrary finite length

Term	**Examples**: (add additional examples from class)
set unordered collection of elements *Equal means agree on membership of all elements*	$7 \in \{43, 7, 9\}$ $\quad 2 \notin \{43, 7, 9\}$
n-**tuple** ordered sequence of elements with n "slots" *Equal means corresponding components equal*	
string ordered finite sequence of elements each from specified set *Equal means same length and corresponding characters equal*	

RNA strands as strings



Each RNA strand is a **string** whose symbols are elements of the set $B = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\}$.

Definition by recursion

New! Recursive Definitions of Sets: The set S (pick a name) is defined by:

Basis Step:	Specify finitely many elements of S
Recursive Step:	Give a rule for creating a new element of S from known values existing in S, and potentially other values.

Definition The set of RNA strands S is defined (recursively) by:

Basis Step:	$\texttt{A} \in S, \texttt{C} \in S, \texttt{U} \in S, \texttt{G} \in S$
Recursive Step:	If $s \in S$ and $b \in B$, then $sb \in S$

Two different RNA strands:

Defining sets

- Roster method
- Set builder notation
- Definition by recursion
- **New** Applying operations to other sets
  - Cartesian product, set-wise concatenation

$B = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\}$

Set	**Example elements in this set:**
Fill in possible set	$(\mathtt{A}, \mathtt{C})$ $(\mathtt{U}, \mathtt{U})$
$B \times \{-1, 0, 1\}$	Fill in example elements
$\{-1, 0, 1\} \times B$	Fill in example elements
Fill in possible set	$(0, 0, 0)$
$\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\} \circ \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\}$	Fill in example elements
Fill in possible set	$\mathtt{GGGG}$

Defining functions

A function is defined by

(1) domain Nonempty set

(2) codomain Nonempty set

(3) rule assigning each element in the domain exactly one element in the codomain Table, formula, etc.

Notation:

Defining functions recursively when domain is recursively defined

Definition (Of a function, recursively) A function $rnalen$ that computes the length of RNA strands in S is defined by:

		$rnalen : S$	$\to \mathbb{Z}^+$
Basis Step:	If $b \in B$ then	$rnalen(b)$	$= 1$
Recursive Step:	If $s \in S$ and $b \in B$, then	$rnalen(sb)$	$= 1 + rnalen(s)$

The domain of $rnalen$ is ____________________. The codomain of $rnalen$ is ____________________.

$rnalen(\texttt{ACU}) =$ ____

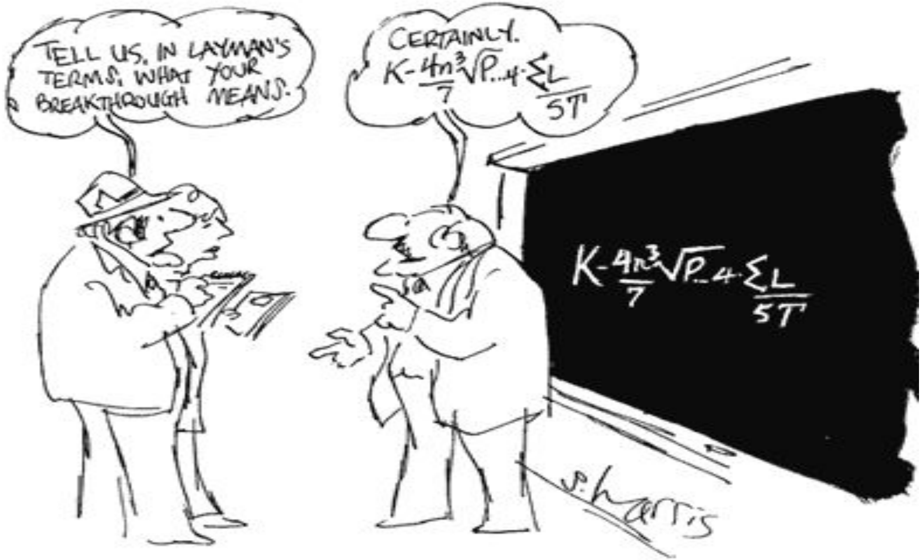
Friday's learning goals

- 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

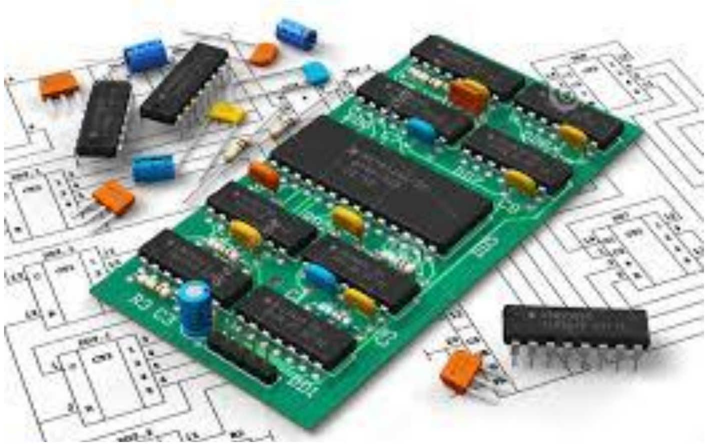
In the past two classes, when have we used numbers?

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 Wednesday's class: the base b expansion of n is a string over the alphabet $\{x \in \mathbb{N} \mid x < b\}$ 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

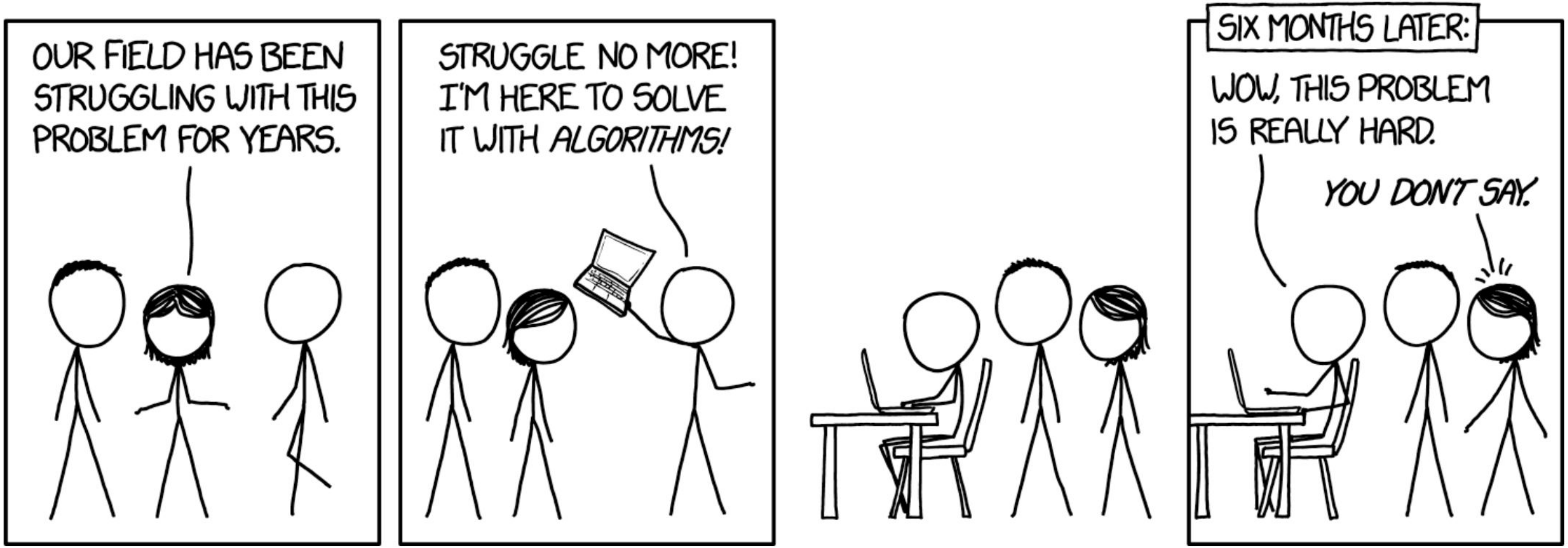
Base b expansion

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

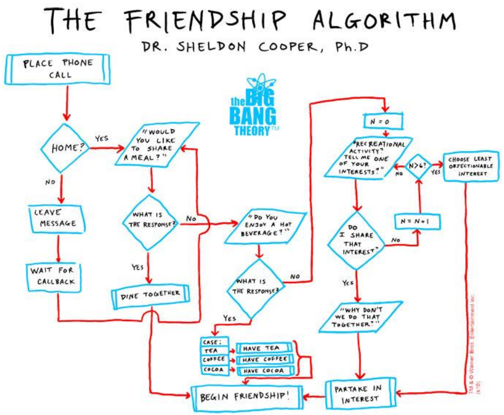


xkcd

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: 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 $0 \le r < d$ and $n = dq + r$. **Notation:** $q = n$ **div** d and $r = n$ **mod** d

When k>1

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

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

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?

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