Week 2 at a glance

We will be learning and practicing to:

- Model systems with tools from discrete mathematics and reason about implications of modelling choices. Explore applications in CS through multiple perspectives, including software, hardware, and theory.
  - Selecting and representing appropriate data types and using notation conventions to clearly communicate choices
  - Determining the properties of positional number representations, including overflow and bit operations
- Translate between different representations to illustrate a concept.
  - Translating between symbolic and English versions of statements using precise mathematical language
  - Tracing algorithms specified in pseudocode
  - Representing numbers using positional representations, including decimal, binary, hexadecimal, fixed-width representations, and 2s complement
- Use precise notation to encode meaning and present arguments concisely and clearly
  - Precisely describing a set using appropriate notation e.g. roster method, set builder notation, and recursive definitions
  - Defining functions using multiple representations
- Know, select and apply appropriate computing knowledge and problem-solving techniques. Reason about computation and systems. Use mathematical techniques to solve problems. Determine appropriate conceptual tools to apply to new situations. Know when tools do not apply and try different approaches. Critically analyze and evaluate candidate solutions.
  - Using a recursive definition to evaluate a function or determine membership in a set
  - Using the definitions of the div and mod operators on integers

TODO:

#FinAid Assignment on Canvas (complete as soon as possible)

Review quiz based on class material each day (due Friday April 12, 2024)

Homework assignment 2 (due Tuesday April 16, 2024).

CC BY-NC-SA 2.0 Version April 25, 2024 (1)

Week 2 Monday: Sets, functions, and algorithms

Let's practice with functions related to some of our applications so far.

Recall: We model the collection of user ratings of the four movies Dune, Oppenheimer, Barbie, Nimona as the set $\{-1, 0, 1\}^4$. One function that compares pairs of ratings is

$$d_0 : \{-1, 0, 1\}^4 \times \{-1, 0, 1\}^4 \to \mathbb{R}$$

given by

$$d_0(\ (\ (x_1, x_2, x_3, x_4), (y_1, y_2, y_3, y_4)\ )\ ) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2 + (x_4 - y_4)^2}$$

Notice: any ordered pair of ratings is an okay input to d_0.

Notice: there are (at most)

$$(3 \cdot 3 \cdot 3 \cdot 3) \cdot (3 \cdot 3 \cdot 3 \cdot 3) = 3^8 = 6561$$

many pairs of ratings. There are therefore lots and lots of real numbers that are not the output of d_0.

Recall: RNA is made up of strands of four different bases that encode genomic information in specific ways. The bases are elements of the set $B = \{\mathtt{A}, \mathtt{C}, \mathtt{U}, \mathtt{G}\}$. The set of RNA strands S is defined (recursively) by:

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

where sb is string concatenation.

CC BY-NC-SA 2.0 Version April 25, 2024 (2)

Pro-tip: informal definitions sometime use $\cdots$ to indicate "continue the pattern". Often, to make this pattern precise we use recursive definitions.

Name	Domain	Codomain	Rule	Example
rnalen	S	$\mathbb{Z}^+$	Basis Step: If $b \in B$ then $rnalen(b) = 1$ Recursive Step: If $s \in S$ and $b \in B$, then $rnalen(sb) = 1 + rnalen(s)$	$rnalen(\texttt{AC}) \overset{\text{rec step}}{=} 1 + rnalen(\texttt{A})$ $\overset{\text{basis step}}{=} 1 + 1 = 2$
basecount	$S \times B$	$\mathbb{N}$	Basis Step: If $b_1 \in B, b_2 \in B$ then $basecount(\ (b_1, b_2)\ ) =$ $\begin{cases} 1 & \text{when } b_1 = b_2 \\ 0 & \text{when } b_1 \neq b_2 \end{cases}$ Recursive Step: If $s \in S, b_1 \in B, b_2 \in B$ $basecount(\ (sb_1, b_2)\ ) =$ $\begin{cases} 1 + basecount(\ (s, b_2)\ ) & \text{when } b_1 = b_2 \\ basecount(\ (s, b_2)\ ) & \text{when } b_1 \neq b_2 \end{cases}$	$basecount(\ (\texttt{ACU}, \texttt{C})\ ) =$
"2 to the power of"	$\mathbb{N}$	$\mathbb{N}$	Basis Step: $2^0 = 1$ Recursive Step: If $n \in \mathbb{N}, 2^{n+1} =$	
"b to the power of i"	$\mathbb{Z}^+ \times \mathbb{N}$	$\mathbb{N}$	Basis Step: $b^0 = 1$ Recursive Step: If $i \in \mathbb{N}, b^{i+1} = b \cdot b^i$	

$2^0 = 1$	$2^1 = 2$	$2^2 = 4$	$2^3 = 8$	$2^4 = 16$	$2^5 = 32$	$2^6 = 64$	$2^7 = 128$	$2^8 = 256$	$2^9 = 512$	$2^{10} = 1024$

CC BY-NC-SA 2.0 Version April 25, 2024 (3)

Integer division and remainders (aka The Division Algorithm) Let n be an integer and d a positive integer. There are unique integers q and r, with $0 \leq r < d$, such that $n = dq + r$. In this case, d is called the divisor, n is called the dividend, q is called the quotient, and r is called the remainder.

Because these numbers are guaranteed to exist, the following functions are well-defined:

- **div** $: \mathbb{Z} \times \mathbb{Z}^+ \to \mathbb{Z}$ given by **div** $(\ (n, d)\ )$ is the quotient when n is the dividend and d is the divisor.
- **mod** $: \mathbb{Z} \times \mathbb{Z}^+ \to \mathbb{Z}$ given by **mod** $(\ (n, d)\ )$ is the remainder when n is the dividend and d is the divisor.

Because these functions are so important, we sometimes use the notation n **div** $d =$ **div** $(\ (n, d)\ )$ and n **mod** $d =$ **mod** $(\ (n, d)\ )$.

Pro-tip: The functions **div** and **mod** are similar to (but not exactly the same as) the operators / and % in Java and python.

Example calculations:

20 **div** 4

20 **mod** 4

20 **div** 3

20 **mod** 3

-20 **div** 3

-20 **mod** 3

CC BY-NC-SA 2.0 Version April 25, 2024 (4)

Week 2 Wednesday: Representing numbers

Modeling uses data-types that are encoded in a computer. The details of the encoding impact the efficiency of algorithms we use to understand the systems we are modeling and the impacts of these algorithms on the people using the systems. Case study: how to encode numbers?

Definition 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 (symbols for) nonnegative integers less than b, $a_{k-1} \neq 0$, and

$$n = \sum_{i=0}^{k-1} a_i b^i$$

Notice: *The base b expansion of a positive integer n is a string over the alphabet $\{x \in \mathbb{N} \mid x < b\}$ whose leftmost character is nonzero.*

Base b	Collection of possible coefficients in base b expansion of a positive integer
Binary ($b = 2$)	$\{0, 1\}$
Ternary ($b = 3$)	$\{0, 1, 2\}$
Octal ($b = 8$)	$\{0, 1, 2, 3, 4, 5, 6, 7\}$
Decimal ($b = 10$)	$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
Hexadecimal ($b = 16$)	$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$ letter coefficient symbols represent numerical values $(A)_{16} = (10)_{10}$ $(B)_{16} = (11)_{10}$ $(C)_{16} = (12)_{10}$ $(D)_{16} = (13)_{10}$ $(E)_{16} = (14)_{10}$ $(F)_{16} = (15)_{10}$

CC BY-NC-SA 2.0 Version April 25, 2024 (5)

Examples:

$(1401)_2$

$(1401)_{10}$

$(1401)_{16}$

New! An algorithm is a finite sequence of precise instructions for solving a problem.

Algorithms can be expressed in English or in more formalized descriptions like pseudocode or fully executable programs.

Sometimes, we can define algorithms whose output matches the rule for a function we already care about. Consider the (integer) logarithm function

$$logb : \{b \in \mathbb{Z} \mid b > 1\} \times \mathbb{Z}^+ \quad \rightarrow \quad \mathbb{N}$$

defined by

$$logb(\ (b, n)\ ) = \text{greatest integer } y \text{ so that } b^y \text{ is less than or equal to } n$$

Calculating integer part of base b logarithm

```
1   procedure logb(b,n: positive integers with b>1)
2   i := 0
3   while n > b-1
4     i := i+1
5     n := n div b
6   return i {i holds the integer part of the base b logarithm of n}
```

Trace this algorithm with inputs $b = 3$ and $n = 17$

	b	n	i	$n > b - 1$?
Initial value	3	17		
After 1 iteration				
After 2 iterations				
After 3 iterations				

Compare: does the output match the rule for the (integer) logarithm function?

CC BY-NC-SA 2.0 Version April 25, 2024 (6)

Two algorithms for constructing base b expansion from decimal representation

Most significant first: Start with left-most coefficient of expansion (highest value)

Informally: Build up to the value we need to represent in "greedy" approach, using units determined by base.

Calculating base b expansion, from left

```
1  procedure baseb1(n, b: positive integers with b > 1)
2  v := n
3  k := 1+ output of logb algorithm with inputs b and n
4  for i := 1 to k
5     a_{k-i} := 0
6     while v ≥ b^{k-i}
7        a_{k-i} := a_{k-i} + 1
8        v := v - b^{k-i}
9  return (a_{k-1}, ..., a_0){(a_{k-1} ... a_0)_b is the base b expansion of n}
```

CC BY-NC-SA 2.0 Version April 25, 2024 (7)

Least significant first: Start with right-most coefficient of expansion (lowest value)

Idea: (when $k > 1$)

$$\begin{aligned} n &= a_{k-1}b^{k-1} + \cdots + a_1 b + a_0 \\ &= b(a_{k-1}b^{k-2} + \cdots + a_1) + a_0 \end{aligned}$$

so $a_0 = n$ **mod** b and $a_{k-1}b^{k-2} + \cdots + a_1 = n$ **div** b.

Calculating base b expansion, from right

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

CC BY-NC-SA 2.0 Version April 25, 2024 (8)

Week 2 Friday: Algorithms for numbers

Find and fix any and all mistakes with the following:

(a) $(1)_2 = (1)_8$

(b) $(142)_{10} = (142)_{16}$

(c) $(20)_{10} = (10100)_2$

(d) $(35)_8 = (1D)_{16}$

Practice: write an algorithm for converting from base b_1 expansion to base b_2 expansion:

Definition For b an integer greater than 1, w a positive integer, and n a nonnegative integer _____, the **base** b **fixed-width** w **expansion of** n is

$$(a_{w-1} \cdots a_1 a_0)_{b,w}$$

where $a_0, a_1, \ldots, a_{w-1}$ are nonnegative integers less than b and

$$n = \sum_{i=0}^{w-1} a_i b^i$$

Decimal $b = 10$	Binary $b = 2$	Binary fixed-width 10 $b = 2$, $w = 10$	Binary fixed-width 7 $b = 2$, $w = 7$	Binary fixed-width 4 $b = 2$, $w = 4$
$(20)_{10}$				

CC BY-NC-SA 2.0 Version April 25, 2024 (9)

Definition For b an integer greater than 1, w a positive integer, w' a positive integer, and x a real number the **base b fixed-width expansion of x with integer part width w and fractional part width w'** is $(a_{w-1} \cdots a_1 a_0 . c_1 \cdots c_{w'})_{b,w,w'}$ where $a_0, a_1, \ldots, a_{w-1}, c_1, \ldots, c_{w'}$ are nonnegative integers less than b and

$$x \geq \sum_{i=0}^{w-1} a_i b^i + \sum_{j=1}^{w'} c_j b^{-j} \qquad \text{and} \qquad x < \sum_{i=0}^{w-1} a_i b^i + \sum_{j=1}^{w'} c_j b^{-j} + b^{-w'}$$

3.75 in fixed-width binary, integer part width 2, fractional part width 8	
0.1 in fixed-width binary, integer part width 2, fractional part width 8	

```
welcome $jshell
|  Welcome to JShell -- Version 10.0.1
|  For an introduction type: /help intro

jshell> 0.1
$1 ==>

jshell> 0.2
$2 ==>

jshell> 0.1 + 0.2
$3 ==>

jshell> Math.sqrt(2)
$4 ==>

jshell> Math.sqrt(2)*Math.sqrt(2)
$5 ==>

jshell>
```

Note: Java uses floating point, not fixed width representation, but similar rounding errors appear in both.

CC BY-NC-SA 2.0 Version April 25, 2024 (10)

Representing negative integers in binary: Fix a positive integer width for the representation w, $w > 1$.

	To represent a positive integer n	To represent a negative integer $-n$
Sign-magnitude	$[0a_{w-2} \cdots a_0]_{s,w}$, where $n = (a_{w-2} \cdots a_0)_{2,w-1}$ Example $n = 17$, $w = 7$:	$[1a_{w-2} \cdots a_0]_{s,w}$, where $n = (a_{w-2} \cdots a_0)_{2,w-1}$ Example $-n = -17$, $w = 7$:
2s complement	$[0a_{w-2} \cdots a_0]_{2c,w}$, where $n = (a_{w-2} \cdots a_0)_{2,w-1}$ Example $n = 17$, $w = 7$:	$[1a_{w-2} \cdots a_0]_{2c,w}$, where $2^{w-1} - n = (a_{w-2} \cdots a_0)_{2,w-1}$ Example $-n = -17$, $w = 7$:

For positive integer n, to represent $-n$ in 2s complement with width w,

- Calculate $2^{w-1} - n$, convert result to binary fixed-width $w - 1$, pad with leading 1, or
- Express $-n$ as a sum of powers of 2, where the leftmost 2^{w-1} is negative weight, or
- Convert n to binary fixed-width w, flip bits, add 1 (ignore overflow)

Challenge: use definitions to explain why each of these approaches works.

CC BY-NC-SA 2.0 Version April 25, 2024 (11)

Representing 0:

So far, we have representations for positive and negative integers. What about 0?

	To represent a **non-negative** integer n	To represent a **non-positive** integer $-n$
Sign-magnitude	$[0a_{w-2}\cdots a_0]_{s,w}$, where $n = (a_{w-2}\cdots a_0)_{2,w-1}$ Example $n = 0$, $w = 7$:	$[1a_{w-2}\cdots a_0]_{s,w}$, where $n = (a_{w-2}\cdots a_0)_{2,w-1}$ Example $-n = 0$, $w = 7$:
2s complement	$[0a_{w-2}\cdots a_0]_{2c,w}$, where $n = (a_{w-2}\cdots a_0)_{2,w-1}$ Example $n = 0$, $w = 7$:	$[1a_{w-2}\cdots a_0]_{2c,w}$, where $2^{w-1} - n = (a_{w-2}\cdots a_0)_{2,w-1}$ Example $-n = 0$, $w = 7$:

CC BY-NC-SA 2.0 Version April 25, 2024 (12)

Review Quiz

1. Functions and algorithms

   (a) What is a recursive definition of the set $\mathbb{Z}^+ \times \mathbb{N}$ that is the domain of the function "b to the power of i"? For convenience, we'll refer to this set as X in the options below.

   Basis step: $(1, 0) \in X$. Recursive step: If $(m, n) \in X$ then $(m + 1, n + 1) \in X$ too.

   Basis step: $(1, 0) \in X$. Recursive step: If $(m, m - 1) \in X$ then $(m + 1, m) \in X$ too.

   Basis step: $(b, 0) \in X$ for each $b \in \mathbb{Z}^+$. Recursive step: If $(m, n) \in X$ then $(m + 1, n + 1) \in X$ too.

   Basis step: $(b, 0) \in X$ for each $b \in \mathbb{Z}^+$. Recursive step: If $(m, n) \in X$ then $(m, n + 1) \in X$ too.

   None of the above.

   (b) When running the algorithm *logb* for calculating the integer part of base b logarithm with inputs $b = 4$ and $n = 25$, which of the following calculations are helpful? Select all and only the calculations that are both relevant to the algorithm trace **and** are correct.

   25 **div** $4 = 5$
   25 **div** $4 = 6$
   25 **div** $4 = 1$
   4 **div** $25 = 0$
   4 **div** $25 = 5$
   4 **div** $25 = 1$
   6 **div** $4 = 1$
   5 **div** $4 = 1$
   4 **div** $4 = 1$

2. Base expansions

   (a) Give the value (using usual mathematical conventions) of each of the following base expansions.

   i. $(10)_2$
   ii. $(10)_4$
   iii. $(17)_{16}$
   iv. $(211)_3$
   v. $(3)_8$

   (b) Recall the definitions from class for number representations for **base b expansion of n**, **base b fixed-width w expansion of n**, and **base b fixed-width expansion of x with integer part width w and fractional part width w'**.

   For example, the base 2 (binary) expansion of 4 is $(100)_2$ and the base 2 (binary) fixed-width 8 expansion of 4 is $(00000100)_{2,8}$ and the base 2 (binary) fixed-width expansion of 4 with integer part width 3 and fractional part width 2 of 4 is $(100.00)_{2,3,2}$

   Compute the listed expansions. Enter your number using the notation for base expansions with parentheses but without subscripts. For example, if your answer were $(100)_{2,3}$ you would type `(100)2,3` into Gradescope.

   i. Give the binary (base 2) expansion of the number whose octal (base 8) expansion is

   $$(371)_8$$

CC BY-NC-SA 2.0 Version April 25, 2024 (13)

ii. Give the decimal (base 10) expansion of the number whose octal (base 8) expansion is

$$(371)_8$$

iii. Give the octal (base 8) fixed-width 3 expansion of $(9)_{10}$.

iv. Give the ternary (base 3) fixed-width 8 expansion of $(9)_{10}$.

v. Give the hexadecimal (base 16) fixed-width 6 expansion of $(16711935)_{10}$.[1]

vi. Give the hexadecimal (base 16) fixed-width 4 expansion of

$$(1011\ 1010\ 1001\ 0000)_2$$

Note: the spaces between each group of 4 bits above are for your convenience only. How might they help your calculations?

vii. Give the binary fixed width expansion of 0.125 with integer part width 2 and fractional part width 4.

viii. Give the binary fixed width expansion of 1 with integer part width 2 and fractional part width 3.

(c) Select all and only the correct choices below.

i. Suppose you were told that the positive integer n_1 has the property that n_1 **div** $2 = 0$. Which of the following can you conclude?

A. n_1 has a binary (base 2) expansion

B. n_1 has a ternary (base 3) expansion

C. n_1 has a hexadecimal (base 16) expansion

D. n_1 has a base 2 fixed-width 1 expansion

E. n_1 has a base 2 fixed-width 20 expansion

ii. Suppose you were told that the positive integer n_2 has the property that n_2 **mod** $4 = 0$. Which of the following can you conclude?

A. the leftmost symbol in the binary (base 2) expansion of n_2 is 1

B. the leftmost symbol in the base 4 expansion of n_2 is 1

C. the rightmost symbol in the base 4 expansion of n_2 is 0

D. the rightmost symbol in the octal (base 8) expansion of n_2 is 0

(d) Recall the definitions of signed integer representations from class: sign-magnitude and 2s complement.

i. Give the 2s complement width 6 representation of the number represented in binary fixed-width 5 representation as $(00101)_{2,5}$.

ii. Give the 2s complement width 6 representation of the number represented in binary fixed-width 5 representation as $(10101)_{2,5}$.

iii. Give the 2s complement width 4 representation of the number represented in sign-magnitude width 4 as $[1111]_{s,4}$.

iv. Give the sign magnitude width 4 representation of the number represented in 2s complement width 4 as $[1111]_{2c,4}$.

v. Give the sign magnitude width 6 representation of the number represented in sign magnitude width 4 as $[1111]_{s,4}$.

[1] This matches a frequent debugging task – sometimes a program will show a number formatted as a base 10 integer that is much better understood with another representation.

vi. Give the 2s complement width 6 representation of the number represented in 2s complement width 4 as $[1111]_{2c,4}$.

3. Multiple representations

We saw last week that, mathematically, a color can be represented as a 3-tuple (r, g, b) where r represents the red component, g the green component, b the blue component and where each of r, g, b must be from the collection $\{x \in \mathbb{N} \mid 0 \leq x \leq 255\}$. As an alternative representation, in this assignment we'll use base b fixed-width expansions to represent colors as individual numbers.

Definition: A **hex color** is a nonnegative integer, n, that has a base 16 fixed-width 6 expansion

$$n = (r_1 r_2 g_1 g_2 b_1 b_2)_{16,6}$$

where $(r_1 r_2)_{16,2}$ is the red component, $(g_1 g_2)_{16,2}$ is the green component, and $(b_1 b_2)_{16,2}$ is the blue.

(a) What is the hex color corresponding to full black? Namely, this means setting the value in each of the red, green, and blue components to be the minimum 0.

0
$(0, 0, 0)$
$15^5 + 15^4 + 15^3 + 15^2 + 15^1 + 1$
$15 \cdot 16^5 + 15 \cdot 16^4 + 15 \cdot 16^3 + 15 \cdot 16^2 + 15 \cdot 16^1 + 15$
$16^5 + 16^4 + 16^3 + 16^2 + 16^1 + 1$

(b) What is the hex color corresponding to full white? Namely, this means setting the value in each of the red, green, and blue components to be the maximum 255.

0
$(0, 0, 0)$
$15^5 + 15^4 + 15^3 + 15^2 + 15^1 + 1$
$15 \cdot 16^5 + 15 \cdot 16^4 + 15 \cdot 16^3 + 15 \cdot 16^2 + 15 \cdot 16^1 + 15$
$16^5 + 16^4 + 16^3 + 16^2 + 16^1 + 1$

(c) Select all and only correct representations of the hex color which is full green (so the red and blue components are 0 and green is set to 255).

$(00FF00)_{16,6}$
$(00FF00)_{16}$
255
$255 \cdot 256$
$255 \cdot 16^2$
65280

(d) Which of the following is a definition using set builder notation for the set of hex colors. (Select all and only correct choices)

$\{x \in \mathbb{Z} \mid 0 \leq x \leq 16777215\}$
$\{x \in \mathbb{Z} \mid 0 \leq x \leq 16^6 - 1\}$
$\{x \in \mathbb{N} \mid x \leq 16777215\}$
$\{x \in \mathbb{N} \mid x \leq 16^6 - 1\}$

CC BY-NC-SA 2.0 Version April 25, 2024 (15)