Let's get started

We want you to be successful.

We will work together to build an environment in CSE 20 that supports your learning in a way that respects your perspectives, experiences, and identities (including race, ethnicity, heritage, gender, sex, class, sexuality, religion, ability, age, educational background, etc.). Our goal is for you to engage with interesting and challenging concepts and feel comfortable exploring, asking questions, and thriving.

If you or someone you know is suffering from food and/or housing insecurities there are UCSD resources here to help:

Basic Needs Office: https://basicneeds.ucsd.edu/

Triton Food Pantry (in the old Student Center) is free and anonymous, and includes produce:

https://www.facebook.com/tritonfoodpantry/

Mutual Aid UCSD: https://mutualaiducsd.wordpress.com/

Financial aid resources, the possibility of emergency grant funding, and off-campus housing referral resources are available: see your College Dean of Student Affairs.

If you find yourself in an uncomfortable situation, ask for help. We are committed to upholding University policies regarding nondiscrimination, sexual violence and sexual harassment. Here are some campus contacts that could provide this help: Counseling and Psychological Services (CAPS) at 858 534-3755 or http://caps.ucsd.edu; OPHD at 858 534-8298 or ophd@ucsd.edu , http://ophd.ucsd.edu; CARE at Sexual Assault Resource Center at 858 534-5793 or sarc@ucsd.edu , http://care.ucsd.edu.

Please reach out (minnes@ucsd.edu) if you need support with extenuating circumstances affecting CSE 20.

Welcome to CSE 20: Discrete Math for CS in Spring 2024!

Class website: https://canvas.ucsd.edu/

Instructor: Prof. Mia Minnes "Minnes" rhymes with Guinness, minnes@ucsd.edu, http://cseweb.ucsd.edu/ minnes

Our team: One instructor + two TAs and eleven tutors + all of you

Fill in contact info for students around you, if you'd like:

On a typical week in CSE 20: **MWF** Lectures (sometimes with pre-class reading), **W** Discussion, Review quiz, then **T** Homework due. Office hours (hosted by instructors and TAs and tutors) where you can come to talk about course concepts and ask for help as you work through sample problems and Q+A on Piazza available throughout the week. CSE 20 has one project and two tests this quarter. Demonstration of class website on Canvas (https://canvas.ucsd.edu/):

1. Syllabus
2. Notes for class + annotations
3. Assignments (PDF, tex, solutions)
4. Gradescope
5. Piazza
6. Dates

Pro-tip: you can use MATH109 to replace CSE20 for prerequisites and other requirements.

Themes and applications for CSE 20

- **Technical skepticism**: 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.
- **Multiple representations**: Understand, guide, shape impact of computing on society/the world. Connect the role of Theory CS classes to other applications (in undergraduate CS curriculum and beyond). Model problems using appropriate mathematical concepts. Clearly and unambiguously communicate computational ideas using appropriate formalism. Translate across levels of abstraction.

Applications: Numbers (how to represent them and use them in Computer Science), Recommendation systems and their roots in machine learning (with applications like Netflix), "Under the hood" of computers (circuits, pixel color representation, data structures), Codes and information (secret message sharing and error correction), Bioinformatics algorithms and genomics (DNA and RNA).

Week 1 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
- Translate between different representations to illustrate a concept.
  - Translating between symbolic and English versions of statements using precise mathematical language
- Use precise notation to encode meaning and present arguments concisely and clearly
  - Defining important sets of numbers, e.g. set of integers, set of rational numbers
  - 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

TODO:

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

Review quiz based on class material each day (due Friday April 5, 2024)

Homework assignment 1 (due Tuesday April 9, 2024).

Week 1 Monday: Modeling applications

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

- how often they use it
- watch list / search history (specific titles / genre / actors / directors)
- watch time, gender, age, location, language settings, device
- previous recommendation interaction - click / watch trailers
- rating

Privacy / Cost concerns from unnecessary data

In machine learning, clustering can be used to group similar data for prediction and recommendation. For example, each Netflix user's viewing history can be represented as a n-tuple indicating their preferences about movies in the database, where n is the number of movies in the database. People with similar tastes in movies can then be clustered to provide recommendations of movies for one another. Mathematically, clustering is based on a notion of distance between pairs of n-tuples.

Feature (_ , _ , _ , _ , _)

Data Types: sets, n-tuples, and strings

Term	**Examples**: (add additional examples from class)
set unordered collection of elements *repetition doesn't matter* *Equal sets agree on membership of all elements*	$7 \in \{43, 7, 9\}$ $2 \notin \{43, 7, 9\}$ $\{7, 43, 9\} = \{43, 7, 9\} = \{43, 7, 9, 7\}$
n-tuple aka vector ordered sequence of elements with n "slots" ($n > 0$) *repetition matters, fixed length* *Equal n-tuples have corresponding components equal*	3 tuples $(1, 2, 3) \neq (3, 2, 1)$ $(1, 2) \neq (1, 2, 3)$
string ordered finite sequence of elements each from specified set (called the alphabet over which the string is defined) *repetition matters, arbitrary finite length* *Equal strings have same length and corresponding characters equal*	over English Alphabet have string hello ≠ helloo
	over alphabet whose characters are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 have string 20 ≠ 02

Special cases:

When $n = 2$, the 2-tuple is called an **ordered pair**. (_ , _)

A string of length 0 is called the **empty string** and is denoted λ. lambda

A set with no elements is called the **empty set** and is denoted $\{\}$ or $\emptyset$.

In the table below, each row represents a user's ratings of movies: ✓ (check) indicates the person liked the movie, ✗ (x) that they didn't, and • (dot) that they didn't rate it one way or another (neutral rating or didn't watch). Can encode these ratings numerically with 1 for ✓ (check), −1 for ✗ (x), and 0 for • (dot).

Pros: Small # of rating values means less memory needed representing ratings numerically gives access to mathematical operations storing ratings as 4-tuples means can infer movie being rated from the component of 4-tuple.

Cons: Small # of rating values means less nuanced and conflates didn't watch w/ no opinion. restricted to a static database that has 4 prespecified movies so hard to add information about new movies.

Person	Dune	Oppenheimer	Barbie	Nimona	Ratings written as a 4-tuple
P_1	✗	•	✓	✓	(-1, 0, 1, 1)
P_2	✓	✓	✗	✓	(1, 1, -1, 1)
P_3	✓	✓	✓	✓	(1, 1, 1, 1)
P_4	•	✗	✓	✓	
You				✓	

Conclusion: Modeling involves choosing data types to represent and organize data

Week 1 Wednesday: Defining sets

Notation and prerequisites

Term	Notation Example(s)	We say in English . . .
all reals	$\mathbb{R}$	The (set of all) real numbers (numbers on the number line)
all integers	$\mathbb{Z}$	The (set of all) integers (whole numbers including negatives, zero, and positives)
all positive integers	$\mathbb{Z}^+$	The (set of all) strictly positive integers
all natural numbers	$\mathbb{N}$	The (set of all) natural numbers. **Note**: we use the convention that 0 is a natural number.

To define sets:

To define a set using **roster method**, explicitly list its elements. That is, start with { then list elements of the set separated by commas and close with }.

Finite set

{Barbie, Oppenheimer, Dune, Nimona}.

To define a set using **set builder definition**, either form "The set of all x from the universe U such that x is ..." by writing

$$\{x \in U \mid ...x...\}$$

or form "the collection of all outputs of some operation when the input ranges over the universe U" by writing

$$\{...x... \mid x \in U\}$$

We use the symbol $\in$ as "is an element of" to indicate membership in a set.

Example sets: For each of the following, identify whether it's defined using the roster method or set builder notation and give an example element.

Can we infer the data type of the example element from the notation?

$\{-1, 1\}$ roster (not the only) sample element: -1

$\{0, 0\}$ roster sample element: 0

$= \{0\}$

$\{-1, 0, 1\}$ roster (not the only) sample element: 0

$\{(x, x, x) \mid x \in \{-1, 0, 1\}\} = \{(-1,-1,-1), (0,0,0), (1,1,1)\}$. (universe)

ordered 3-tuple. $(1,0,-1)$ is not an element of this set.

$\{\}$ roster empty set, it has no sample elements.

$\{x \in \mathbb{Z}^+ \mid x < 0\} = \{\}$. empty set

$\{x \in \mathbb{Z} \mid x \geq 0\} = \mathbb{N}$

$\{(x, y, z) \mid x \in \{0, 1, -1\}, y \in \{0, 1, -1\}, z \in \{0, 1, -1\}\}$

$= \{(x, y, z) \mid x, y, z \in \{0, 1, -1\}\}$.

$\{x \in \mathbb{Z} \mid x > 0\} = \mathbb{Z}^+$

$\{x \in \mathbb{Z} \mid 1 \leq x \leq 9\} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

$\{\smile, \sun\}$ roster

$\{\mathtt{A}, \mathtt{C}, \mathtt{U}, \mathtt{G}\}$ roster

$\{\mathtt{AUG}, \mathtt{UAG}, \mathtt{UGA}, \mathtt{UAA}\}$ roster

order important
strands are finite length

RNA is made up of strands of four different bases that encode genomic information in specific ways. The bases are elements of the set $B = \{\mathtt{A}, \mathtt{C}, \mathtt{U}, \mathtt{G}\}$. Strands are ordered nonempty finite sequences of bases.

Formally, to define the set of all RNA strands, we need more than roster method or set builder descriptions.

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 rule(s) for creating a new element of S from known values existing in S, and potentially other values.

The set S then consists of all and only elements that are put in S by finitely many (a nonnegative integer number) of applications of the recursive step after the basis step.

Definition The set of nonnegative integers $\mathbb{N}$ is defined (recursively) by:

Basis Step: 0
Recursive Step: when $n \in \mathbb{N}$, $n+1 \in \mathbb{N}$ too.

"when n is an element of $\mathbb{N}$, then so is n+1"

Examples: extra practice

Definition The set of all integers $\mathbb{Z}$ is defined (recursively) by:

Basis Step: 0
Recursive Step: when $x \in \mathbb{Z}$ then $x+1 \in \mathbb{Z}$ and $x-1 \in \mathbb{Z}$.

Examples: extra practice.

practice

Definition 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.

one string next to the other indicates "glue" together to form a new string.

Examples:

ACU $\in S$ guarantees that ACUA $\in S$.

Definition The set of bitstrings (strings of 0s and 1s) is defined (recursively) by:

Basis Step: λ is a bitstring
Recursive Step: If w is a bitstring then so is w0 and so is w1

practice

Notation: We call the set of bitstrings $\{0,1\}^*$ and we say this is the set of all strings over $\{0,1\}$.

Examples:

To define a set we can use the roster method, set builder notation, a recursive definition, and also we can apply a set operation to other sets.

New! Cartesian product of sets and **set-wise concatenation of sets of strings**

Definition: Let X and Y be sets. The **Cartesian product** of X and Y, denoted $X \times Y$, is the set of all ordered pairs (x, y) where $x \in X$ and $y \in Y$

$$X \times Y = \{(x, y) \mid x \in X \text{ and } y \in Y\}$$

Conventions: (1) Cartesian products can be chained together to result in sets of n-tuples and (2) When we form the Cartesian product of a set with itself $X \times X$ we can denote that set as X^2, or X^n for the Cartesian product of a set with itself n times for a positive integer n.

Definition: Let X and Y be sets of strings over the same alphabet. The **set-wise concatenation** of X and Y, denoted $X \circ Y$, is the set of all results of string concatenation xy where $x \in X$ and $y \in Y$

$$X \circ Y = \{xy \mid x \in X \text{ and } y \in Y\}$$

Pro-tip: the meaning of writing one element next to another like xy depends on the data-types of x and y. When x and y are strings, the convention is that xy is the result of string concatenation. When x and y are numbers, the convention is that xy is the result of multiplication. This is (one of the many reasons) why is it very important to declare the data-type of variables before we use them.

Fill in the missing entries in the table: Many have multiple possible correct choices

Set	Example elements in this set and their data type:
B defined on page 8	A C G U
$B \times B$	(A, C) (U, U)
$B \times \{-1, 0, 1\}$	(A, 0)
$\{-1, 0, 1\} \times B$	(0, A)
$\{(0, 0, 0)\}$	$(0, 0, 0)$
$\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\} \circ \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\}$	AC
$\{A, G\} \circ \{GGG\}$	GGGG

Week 1 Friday: Defining functions

Term	Notation Example(s)	We say in English ...
sequence	$x_1, \ldots, x_n$	A sequence x_1 to x_n
summation	$\sum_{i=1}^{n} x_i$ or $\sum_{i=1}^{n} x_i$	The sum of the terms of the sequence x_1 to x_n
piecewise rule definition	$f(x) = \begin{cases} \text{rule 1 for } x & \text{when COND 1} \\ \text{rule 2 for } x & \text{when COND 2} \end{cases}$	Define f of x to be the result of applying rule 1 to x when condition COND 1 is true and the result of applying rule 2 to x when condition COND 2 is true. This can be generalized to having more than two conditions (or cases).
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
absolute value	$\lvert -3 \rvert$	The absolute value of -3
square root	$\sqrt{9}$	The non-negative square root of 9

Pro-tip: the meaning of two vertical lines | | depends on the data-types of what's between the lines. For example, when placed around a number, the two vertical lines represent absolute value. We've seen a single vertial line | used as part of set builder definitions to represent "such that". Again, this is (one of the many reasons) why is it very important to declare the data-type of variables before we use them.

New! Defining functions A function is defined by its (1) domain, (2) codomain, and (3) rule assigning each element in the domain exactly one element in the codomain.

well-defined

The domain and codomain are nonempty sets. collections of objects

The rule can be depicted as a table, formula, piecewise definition, or English description.

The notation is

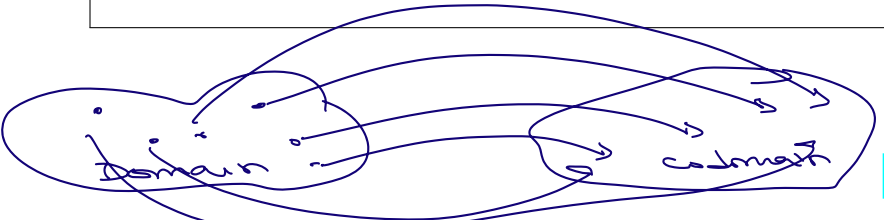
signature

"Let the function FUNCTION-NAME: DOMAIN → CODOMAIN be given by
FUNCTION-NAME(x) = ... for every $x \in DOMAIN$".

rule

or

"Consider the function FUNCTION-NAME: DOMAIN → CODOMAIN defined as
FUNCTION-NAME(x) = ... for every $x \in DOMAIN$".



Alt function: $\mathbb{R} \to \mathbb{R}$

Example: The absolute value function — not fully specified.

TYPES **Domain** $\mathbb{R}$ — set of possible inputs

TYPES **Codomain** $\{x \in \mathbb{R} \mid x \geq 0\}$ — set of possible outputs.

Rule

the absolute value of x is x if $x \geq 0$ and it's $-x$ if $x < 0$

$$|x| = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{if } x < 0 \end{cases}$$

Recall our representation of Netflix users' ratings of movies as n-tuples, where n is the number of movies in the database. Each component of the n-tuple is -1 (didn't like the movie), 0 (neutral rating or didn't watch the movie), or 1 (liked the movie). ($n = 4$ (4 movies))

Consider the ratings $P_1 = (-1, 0, 1, 0)$, $P_2 = (1, 1, -1, 0)$, $P_3 = (1, 1, 1, 0)$, $P_4 = (0, -1, 1, 0)$

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

One approach to answer this question: use **functions** to quantify difference among user preferences.

For example, consider the function (name of function) $d_0 : \{-1, 0, 1\}^4 \times \{-1, 0, 1\}^4 \to \mathbb{R}$ (domain; codomain) given by

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

(input; output)

Using mathematical operations like subtraction, addition, multiplication, square root.

function application.

$$d_0(\ (P_1, P_4)\ ) = \sqrt{(-1)^2 + (+1)^2 + 0^2 + 0^2} = \sqrt{2}$$

input

$d_0(\ (P_2, P_4)\ ) =$

$d_0(\ (P_3, P_4)\ ) =$

Sets defined by roster, set builder, recursive definition, set operations

When the domain of a function is a *recursively defined set*, the rule assigning images to domain elements (outputs) can also be defined recursively.

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

A, C, U, G

AA, CA, UA, GA,
AC, CC, UC, GC,
- - -

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.

informal idea

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

input

$$rnalen : S \to \mathbb{Z}^+$$

Basis Step:	If $b \in B$ then	$rnalen(b) = 1$
Recursive Step:	If $s \in S$ and $b \in B$, then	$rnalen(sb) = 1 + rnalen(s)$

Rule
Rule.

input

The domain of *rnalen* is S

The codomain of *rnalen* is $\mathbb{Z}^+$

Example function application:

$rnalen(\mathtt{ACU}) =$ $1 + rnalen(\mathtt{AC})$ recursive step

AC · U
s b

recursive step.

$= 1 + (1 + rnalen(\mathtt{A}))$

$= 1 + (1 + 1)$ basis step

$= 3$.

Example: A function *basecount* that computes the number of a given base b appearing in a RNA strand s is defined recursively:

$$basecount : S \times B \to \mathbb{N}$$

Basis Step: If $b_1 \in B, b_2 \in B$

$$basecount((b_1, b_2)) = \begin{cases} 1 & \text{when } b_1 = b_2 \\ 0 & \text{when } b_1 \neq b_2 \end{cases}$$

Recursive Step: If $s \in S, b_1 \in B, b_2 \in B$

$$basecount((sb_1, b_2)) = \begin{cases} 1 + basecount((s, b_2)) & \text{when } b_1 = b_2 \\ basecount((s, b_2)) & \text{when } b_1 \neq b_2 \end{cases}$$

Review Quiz

1. Modeling

   (a) Using the example movie database from class with the four movies Dune, Oppenheimer, Barbie, and Nimona which of the following is a 4-tuple that represents the ratings of a user who liked Dune? (Select all and only that apply.)

   i. 1
   ii. $(1, 0, 0)$
   iii. $[1, 1, 1, 0]$
   iv. $\{-1, 0, 0, 1\}$
   v. $(1, -1, 0, 1)$
   vi. $(0, 1, 1, 1)$
   vii. $(1, 1, 1, 1)$

   (b) Using the example movie database from class with the four movies Dune, Oppenheimer, Barbie, and Nimona how many distinct (different) 4-tuples of ratings are there?

   (c) Colors can be described as amounts of red, green, and blue mixed together [1] 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 a value from this collection of numbers:

   {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255}

   Select all and only the true statements below.

   i. $(1, 3, 4)$ fits the definition of a color above.
   ii. $(1, 100, 200, 0)$ fits the definition of a color above.
   iii. $(510, 255)$ fits the definition of a color above.
   iv. There is a color (r_1, g_1, b_1) where $r_1 + g_1 + b_1$ is greater than 765.
   v. There is a color (r_2, g_2, b_2) where $r_2 + g_2 + b_2$ is equal to 1.
   vi. Another way to write the collection of allowed values for red, green, and blue components is

   $$\{x \in \mathbb{N} \mid 0 \leq x \leq 255\}$$

   .

[1] This RGB representation is common in web applications. Many online tools are available to play around with mixing these colors, e.g. `https://www.w3schools.com/colors/colors_rgb.asp`.

vii. Another way to write the collection of allowed values for red, green, and blue components is

$$\{n \in \mathbb{Z} \mid 0 \leq n \leq 255\}$$

.

viii. Another way to write the collection of allowed values for red, green, and blue components is

$$\{y \in \mathbb{Z} \mid -1 < y \leq 255\}$$

.

(d) In the definition of colors as amounts of red, green, and blue mixed together, why are 3-tuples a convenient data structure to use rather than sets or strings?
(Select all and only relevant choices)

i. Ordering matters in n-tuples, so we can use the different components of the 3-tuple to represent the amounts of specific colors.

ii. There are many possible values for each color amount and we don't have individual characters for each value so a string could get unwieldy.

iii. It's possible to have the same value of two or all of the colors, and repetition matters in n-tuples.

2. Sets and functions

(a) Sets are unordered collections. In class, we saw some examples of sets and also how to define sets using roster method and set builder notation.

i. Select all and only the sets below that have 0 as an element.

A. $\{-1, 1\}$
B. $\{0, 0\}$
C. $\{-1, 0, 1\}$
D. $\mathbb{Z}$
E. $\mathbb{Z}^+$
F. $\mathbb{N}$

ii. Select all and only the sets below that have the ordered pair $(2, 0)$ as an element.

A. $\{x \mid x \in \mathbb{N}\}$
B. $\{(x, x) \mid x \in \mathbb{N}\}$
C. $\{(x, x-2) \mid x \in \mathbb{N}\}$
D. $\{(x, y) \mid x \in \mathbb{Z}^+, y \in \mathbb{Z}\}$

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

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	$rnalen(sb)$	$= 1 + rnalen(s)$

i. How many distinct elements are in the set described using set builder notation as

$$\{x \in S \mid rnalen(x) = 1\} \qquad ?$$

ii. How many distinct elements are in the set described using set builder notation as

$$\{x \in S \mid rnalen(x) = 2\} \qquad ?$$

iii. How many distinct elements are in the set described using set builder notation as

$$\{rnalen(x) \mid x \in S \text{ and } rnalen(x) = 2\} \qquad ?$$

iv. How many distinct elements are in the set obtained as the result of the set-wise concatenation $\{\mathtt{AA}, \mathtt{AC}\} \circ \{\mathtt{U}, \mathtt{AA}\}$?

v. How many distinct elements are in the set obtained as the result of the Cartesian product $\{\mathtt{AA}, \mathtt{AC}\} \times \{\mathtt{U}, \mathtt{AA}\}$?

vi. **True** or **False**: There is an example of an RNA strand that is both in the set obtained as the result of the set-wise concatenation $\{\mathtt{AA}, \mathtt{AC}\} \circ \{\mathtt{U}, \mathtt{AA}\}$ and in the set obtained as the result of the Cartesian product $\{\mathtt{AA}, \mathtt{AC}\} \times \{\mathtt{UA}, \mathtt{AA}\}$

Bonus - not for credit: Describe each of the sets above using roster method.

(c) Recall the function which takes an ordered pair of ratings 4-tuples and returns a measure of the difference between them $d_0 : \{-1, 0, 1\}^4 \times \{-1, 0, 1\}^4 \to \mathbb{R}$ given by

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

Consider the function application

$$d_0(\ (\ (-1, 1, 1, 0), (1, 0, -1, 0)\ )\ )$$

i. What is the input?

ii. What is the output?

