

Welcome to the class!

## Themes for CSE 20

- Technical skepticism
- Multiple representations

## Recurring applications in CSE 20

- Clustering and recommendation systems (machine learning, Netflix)
- Genomics and bioinformatics (DNA and RNA)
- Codes and information (secret message sharing and error correction)
- “Under the hood” of computers (circuits, pixel color representation, data structures)

# Friday September 24

| Term                       | Notation Example(s)  | We say in English   |
|----------------------------|--|---|
| $n$ -tuple                 | $(x_1, x_2, x_3)$<br>$(3, 4)$  | The 3-tuple of $x_1$ , $x_2$ , and $x_3$<br>The 2-tuple or <b>ordered pair</b> of 3 and 4   |
| sequence                   | $x_1, \dots, x_n$<br>$x_1, \dots, x_n$ where $n = 0$<br>$x_1, \dots, x_n$ where $n = 1$<br>$x_1, \dots, x_n$ where $n = 2$<br>$x_1, x_2$ | A sequence $x_1$ to $x_n$<br>An empty sequence<br>A sequence containing just $x_1$<br>A sequence containing just $x_1$ and $x_2$ in order<br>A sequence containing just $x_1$ and $x_2$ in order                                      |
| set                        |  | Unordered collection of objects. The set of ...   |
| 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. <b>Note:</b> we use the convention that 0 is a natural number.  |
| roster method              | $\{43, 7, 9\}$<br>$\{9, \mathbb{N}\}$  | The set whose elements are 43, 7, and 9<br>The set whose elements are 9 and $\mathbb{N}$  |
| set builder notation       | $\{x \in \mathbb{Z} \mid x > 0\}$<br>$\{3x \mid x \in \mathbb{Z}\}$  | The set of all $x$ from the integers such that $x$ is greater than 0<br>The set of all integer multiples of 3 <b>Note:</b> we use the convention that writing two numbers next to each other means multiplication.                    |
| function definition        | $f(x) = x + 4$   | Define $f$ of $x$ to be $x + 4$   |
| function application       | $f(7)$<br>$f(z)$<br>$f(g(z))$  | $f$ of 7 <b>or</b> $f$ applied to 7 <b>or</b> the image of 7 under $f$<br>$f$ of $z$ <b>or</b> $f$ applied to $z$ <b>or</b> the image of $z$ under $f$<br>$f$ of $g$ of $z$ <b>or</b> $f$ applied to the result of $g$ applied to $z$ |
| absolute value             | $ -3 $   | The absolute value of $-3$  |
| square root                | $\sqrt{9}$   | The non-negative square root of 9   |
| summation notation         | $\sum_{i=1}^n i$<br>$\sum_{i=1}^n i^2 - 1$   | The sum of the integers from 1 to $n$ , inclusive<br>The sum of $i^2 - 1$ ( $i$ squared minus 1) for each $i$ from 1 to $n$ , inclusive   |
| quotient, integer division | $n \text{ div } m$   | The (integer) quotient upon dividing $n$ by $m$ ; informally: divide and then drop the fractional part  |
| modulo, remainder          | $n \text{ mod } m$   | The remainder upon dividing $n$ by $m$  |

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

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.

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).

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

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 define distance between user preferences.

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

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

*Extra example:* A new movie is released, and  $P_1$  and  $P_2$  watch it before  $P_3$ , and give it ratings;  $P_1$  gives ✓ and  $P_2$  gives ✗. Should this movie be recommended to  $P_3$ ? Why or why not?

*Extra example:* Define the new functions that would be used to compare the 4-tuples of ratings encoding movie preferences now that there are four movies in the database.

# Monday September 27

| Term  | Examples:<br>(add additional examples from class) |
|---|---|
| <b>set</b><br>unordered collection of elements<br><i>Equal means agree on membership of all elements</i>  | $7 \in \{43, 7, 9\}$ $2 \notin \{43, 7, 9\}$      |
| <b><math>n</math>-tuple</b><br>ordered sequence of elements with $n$ “slots”<br><i>Equal means corresponding components equal</i>                 |   |
| <b>string</b><br>ordered finite sequence of elements each from specified set<br><i>Equal means same length and corresponding characters equal</i> |   |

$$\{-1, 1\} \quad \{0, 0\} \quad \{-1, 0, 1\} \quad \mathbb{Z} \quad \mathbb{N} = \{x \in \mathbb{Z} \mid x \geq 0\} \quad \emptyset \quad \mathbb{Z}^+ = \{x \in \mathbb{Z} \mid x > 0\}$$

Which of the sets above are defined using the roster method? Which are defined using set builder notation?

Which of the sets above have 0 as an element?

Can you write any of the sets above more simply?

RNA is made up of strands of four different bases that match up in specific ways.  
The bases are elements of the set  $B = \{\mathbf{A}, \mathbf{C}, \mathbf{G}, \mathbf{U}\}$ .

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

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

where  $sb$  is string concatenation.

Examples:

To define a set we can use the **roster method**, the **set builder notation**, and also ...

**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.

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.

**Wednesday September 29**