

## Monday November 29

*Recall:*

A relation is an **equivalence relation** means it is reflexive, symmetric, and transitive.

An **equivalence class** of an element  $a \in A$  with respect to an equivalence relation  $R$  on the set  $A$  is the set

$$\{s \in A \mid (a, s) \in R\}$$

We write  $[a]_R$  for this set, which is the equivalence class of  $a$  with respect to  $R$ .

A **partition** of a set  $A$  is a set of non-empty, disjoint subsets  $A_1, A_2, \dots, A_n$  such that

$$A = \bigcup_{i=1}^n A_i = \{x \mid \exists i(x \in A_i)\}$$

**Claim:** For each  $a \in U$ ,  $[a]_E \neq \emptyset$ .

**Proof:** Towards a \_\_\_\_\_ consider an arbitrary element  $a$  in  $U$ . We will work to show that  $[a]_E \neq \emptyset$ , namely that  $\exists x \in [a]_E$ . By definition of equivalence classes, we can rewrite this goal as

$$\exists x \in U ( (a, x) \in E )$$

Towards a \_\_\_\_\_, consider  $x = a$ , an element of  $U$  by definition. By \_\_\_\_\_ of  $E$ , we know that  $(a, a) \in E$  and thus the existential quantification has been proved.

**Claim:** For each  $a \in U$ , there is some  $b \in U$  such that  $a \in [b]_E$ .

Towards a \_\_\_\_\_ consider an arbitrary element  $a$  in  $U$ . By definition of equivalence classes, we can rewrite the goal as

$$\exists b \in U ( (b, a) \in E )$$

Towards a \_\_\_\_\_, consider  $b = a$ , an element of  $U$  by definition. By \_\_\_\_\_ of  $E$ , we know that  $(a, a) \in E$  and thus the existential quantification has been proved.

**Claim:** For each  $a, b \in U$ ,  $((a, b) \in E \rightarrow [a]_E = [b]_E)$  and  $((a, b) \notin E \rightarrow [a]_E \cap [b]_E = \emptyset)$

**Corollary:** Given an equivalence relation  $E$  on set  $U$ ,  $\{[x]_E \mid x \in U\}$  is a partition of  $U$ .

Last time, we saw that partitions associated to equivalence relations were useful in the context of modular arithmetic. Today we'll look at a different application.

Recall that in a movie recommendation system, each user's ratings of movies is represented as a  $n$ -tuple (with the positive integer  $n$  being the number of movies in the database), and each component of the  $n$ -tuple is an element of the collection  $\{-1, 0, 1\}$ .

We call  $Rt_5$  the set of all ratings 5-tuples.

Define  $d : Rt_5 \times Rt_5 \rightarrow \mathbb{N}$  by

$$d( ( (x_1, x_2, x_3, x_4, x_5), (y_1, y_2, y_3, y_4, y_5) ) ) = \sum_{i=1}^5 |x_i - y_i|$$

Consider the following binary relations on  $Rt_5$ .

$$E_{proj} = \{ ( (x_1, x_2, x_3, x_4, x_5), (y_1, y_2, y_3, y_4, y_5) ) \in Rt_5 \times Rt_5 \mid (x_1 = y_1) \wedge (x_2 = y_2) \wedge (x_3 = y_3) \}$$

Example ordered pair in  $E_{proj}$ :

Reflexive? Symmetric? Transitive? Antisymmetric?

$$E_{dist} = \{ (u, v) \in Rt_5 \times Rt_5 \mid d( (u, v) ) \leq 2 \}$$

Example ordered pair in  $E_{dist}$ :

Reflexive? Symmetric? Transitive? Antisymmetric?

$$E_{circ} = \{(u, v) \in Rt_5 \times Rt_5 \mid d((0, 0, 0, 0, 0), u) = d((0, 0, 0, 0, 0), v)\}$$

Example ordered pair in  $E_{circ}$ :

Reflexive? Symmetric? Transitive? Antisymmetric?

The partition of  $Rt_5$  defined by \_\_\_\_\_ is

{ { (-1, -1, -1, -1, -1), (-1, -1, -1, -1, 0), (-1, -1, -1, -1, 1), (-1, -1, -1, 0, -1), (-1, -1, -1, 0, 0), (-1, -1, -1, 0, 1), (-1, -1, -1, 1, -1), (-1, -1, -1, 1, 0), (-1, -1, -1, 1, 1) },  
 { (-1, -1, 0, -1, -1), (-1, -1, 0, -1, 0), (-1, -1, 0, -1, 1), (-1, -1, 0, 0, -1), (-1, -1, 0, 0, 0), (-1, -1, 0, 0, 1), (-1, -1, 0, 1, -1), (-1, -1, 0, 1, 0), (-1, -1, 0, 1, 1) },  
 { (-1, -1, 1, -1, -1), (-1, -1, 1, -1, 0), (-1, -1, 1, -1, 1), (-1, -1, 1, 0, -1), (-1, -1, 1, 0, 0), (-1, -1, 1, 0, 1), (-1, -1, 1, 1, -1), (-1, -1, 1, 1, 0), (-1, -1, 1, 1, 1) },  
 { (-1, 0, -1, -1, -1), (-1, 0, -1, -1, 0), (-1, 0, -1, -1, 1), (-1, 0, -1, 0, -1), (-1, 0, -1, 0, 0), (-1, 0, -1, 0, 1), (-1, 0, -1, 1, -1), (-1, 0, -1, 1, 0), (-1, 0, -1, 1, 1) },  
 { (-1, 0, 0, -1, -1), (-1, 0, 0, -1, 0), (-1, 0, 0, -1, 1), (-1, 0, 0, 0, -1), (-1, 0, 0, 0, 0), (-1, 0, 0, 0, 1), (-1, 0, 0, 1, -1), (-1, 0, 0, 1, 0), (-1, 0, 0, 1, 1) },  
 { (-1, 0, 1, -1, -1), (-1, 0, 1, -1, 0), (-1, 0, 1, -1, 1), (-1, 0, 1, 0, -1), (-1, 0, 1, 0, 0), (-1, 0, 1, 0, 1), (-1, 0, 1, 1, -1), (-1, 0, 1, 1, 0), (-1, 0, 1, 1, 1) },  
 { (-1, 1, -1, -1, -1), (-1, 1, -1, -1, 0), (-1, 1, -1, -1, 1), (-1, 1, -1, 0, -1), (-1, 1, -1, 0, 0), (-1, 1, -1, 0, 1), (-1, 1, -1, 1, -1), (-1, 1, -1, 1, 0), (-1, 1, -1, 1, 1) },  
 { (-1, 1, 0, -1, -1), (-1, 1, 0, -1, 0), (-1, 1, 0, -1, 1), (-1, 1, 0, 0, -1), (-1, 1, 0, 0, 0), (-1, 1, 0, 0, 1), (-1, 1, 0, 1, -1), (-1, 1, 0, 1, 0), (-1, 1, 0, 1, 1) },  
 { (-1, 1, 1, -1, -1), (-1, 1, 1, -1, 0), (-1, 1, 1, -1, 1), (-1, 1, 1, 0, -1), (-1, 1, 1, 0, 0), (-1, 1, 1, 0, 1), (-1, 1, 1, 1, -1), (-1, 1, 1, 1, 0), (-1, 1, 1, 1, 1) },  
 { (0, -1, -1, -1, -1), (0, -1, -1, -1, 0), (0, -1, -1, -1, 1), (0, -1, -1, 0, -1), (0, -1, -1, 0, 0), (0, -1, -1, 0, 1), (0, -1, -1, 1, -1), (0, -1, -1, 1, 0), (0, -1, -1, 1, 1) },  
 { (0, -1, 0, -1, -1), (0, -1, 0, -1, 0), (0, -1, 0, -1, 1), (0, -1, 0, 0, -1), (0, -1, 0, 0, 0), (0, -1, 0, 0, 1), (0, -1, 0, 1, -1), (0, -1, 0, 1, 0), (0, -1, 0, 1, 1) },  
 { (0, -1, 1, -1, -1), (0, -1, 1, -1, 0), (0, -1, 1, -1, 1), (0, -1, 1, 0, -1), (0, -1, 1, 0, 0), (0, -1, 1, 0, 1), (0, -1, 1, 1, -1), (0, -1, 1, 1, 0), (0, -1, 1, 1, 1) },  
 { (0, 0, -1, -1, -1), (0, 0, -1, -1, 0), (0, 0, -1, -1, 1), (0, 0, -1, 0, -1), (0, 0, -1, 0, 0), (0, 0, -1, 0, 1), (0, 0, -1, 1, -1), (0, 0, -1, 1, 0), (0, 0, -1, 1, 1) },  
 { (0, 0, 0, -1, -1), (0, 0, 0, -1, 0), (0, 0, 0, -1, 1), (0, 0, 0, 0, -1), (0, 0, 0, 0, 0), (0, 0, 0, 0, 1), (0, 0, 0, 1, -1), (0, 0, 0, 1, 0), (0, 0, 0, 1, 1) },  
 { (0, 0, 1, -1, -1), (0, 0, 1, -1, 0), (0, 0, 1, -1, 1), (0, 0, 1, 0, -1), (0, 0, 1, 0, 0), (0, 0, 1, 0, 1), (0, 0, 1, 1, -1), (0, 0, 1, 1, 0), (0, 0, 1, 1, 1) },  
 { (0, 1, -1, -1, -1), (0, 1, -1, -1, 0), (0, 1, -1, -1, 1), (0, 1, -1, 0, -1), (0, 1, -1, 0, 0), (0, 1, -1, 0, 1), (0, 1, -1, 1, -1), (0, 1, -1, 1, 0), (0, 1, -1, 1, 1) },  
 { (0, 1, 0, -1, -1), (0, 1, 0, -1, 0), (0, 1, 0, -1, 1), (0, 1, 0, 0, -1), (0, 1, 0, 0, 0), (0, 1, 0, 0, 1), (0, 1, 0, 1, -1), (0, 1, 0, 1, 0), (0, 1, 0, 1, 1) },  
 { (0, 1, 1, -1, -1), (0, 1, 1, -1, 0), (0, 1, 1, -1, 1), (0, 1, 1, 0, -1), (0, 1, 1, 0, 0), (0, 1, 1, 0, 1), (0, 1, 1, 1, -1), (0, 1, 1, 1, 0), (0, 1, 1, 1, 1) },  
 { (1, -1, -1, -1, -1), (1, -1, -1, -1, 0), (1, -1, -1, -1, 1), (1, -1, -1, 0, -1), (1, -1, -1, 0, 0), (1, -1, -1, 0, 1), (1, -1, -1, 1, -1), (1, -1, -1, 1, 0), (1, -1, -1, 1, 1) },  
 { (1, -1, 0, -1, -1), (1, -1, 0, -1, 0), (1, -1, 0, -1, 1), (1, -1, 0, 0, -1), (1, -1, 0, 0, 0), (1, -1, 0, 0, 1), (1, -1, 0, 1, -1), (1, -1, 0, 1, 0), (1, -1, 0, 1, 1) },  
 { (1, -1, 1, -1, -1), (1, -1, 1, -1, 0), (1, -1, 1, -1, 1), (1, -1, 1, 0, -1), (1, -1, 1, 0, 0), (1, -1, 1, 0, 1), (1, -1, 1, 1, -1), (1, -1, 1, 1, 0), (1, -1, 1, 1, 1) },  
 { (1, 0, -1, -1, -1), (1, 0, -1, -1, 0), (1, 0, -1, -1, 1), (1, 0, -1, 0, -1), (1, 0, -1, 0, 0), (1, 0, -1, 0, 1), (1, 0, -1, 1, -1), (1, 0, -1, 1, 0), (1, 0, -1, 1, 1) },  
 { (1, 0, 0, -1, -1), (1, 0, 0, -1, 0), (1, 0, 0, -1, 1), (1, 0, 0, 0, -1), (1, 0, 0, 0, 0), (1, 0, 0, 0, 1), (1, 0, 0, 1, -1), (1, 0, 0, 1, 0), (1, 0, 0, 1, 1) },  
 { (1, 0, 1, -1, -1), (1, 0, 1, -1, 0), (1, 0, 1, -1, 1), (1, 0, 1, 0, -1), (1, 0, 1, 0, 0), (1, 0, 1, 0, 1), (1, 0, 1, 1, -1), (1, 0, 1, 1, 0), (1, 0, 1, 1, 1) },  
 { (1, 1, -1, -1, -1), (1, 1, -1, -1, 0), (1, 1, -1, -1, 1), (1, 1, -1, 0, -1), (1, 1, -1, 0, 0), (1, 1, -1, 0, 1), (1, 1, -1, 1, -1), (1, 1, -1, 1, 0), (1, 1, -1, 1, 1) },  
 { (1, 1, 0, -1, -1), (1, 1, 0, -1, 0), (1, 1, 0, -1, 1), (1, 1, 0, 0, -1), (1, 1, 0, 0, 0), (1, 1, 0, 0, 1), (1, 1, 0, 1, -1), (1, 1, 0, 1, 0), (1, 1, 0, 1, 1) },  
 { (1, 1, 1, -1, -1), (1, 1, 1, -1, 0), (1, 1, 1, -1, 1), (1, 1, 1, 0, -1), (1, 1, 1, 0, 0), (1, 1, 1, 0, 1), (1, 1, 1, 1, -1), (1, 1, 1, 1, 0), (1, 1, 1, 1, 1) } }

The partition of  $Rt_5$  defined by  $E =$  \_\_\_\_\_ is

{  
 [ (0, 0, 0, 0, 0) ]<sub>E</sub>  
 , [ (0, 0, 0, 0, 1) ]<sub>E</sub>  
 , [ (0, 0, 0, 1, 1) ]<sub>E</sub>  
 , [ (0, 0, 1, 1, 1) ]<sub>E</sub>  
 , [ (0, 1, 1, 1, 1) ]<sub>E</sub>  
 , [ (1, 1, 1, 1, 1) ]<sub>E</sub>  
 }

How many elements are in each part of the partition?

# Review

1. Select all and only the partitions of  $\{1, 2, 3, 4, 5\}$  from the sets below.

- (a)  $\{1, 2, 3, 4, 5\}$
- (b)  $\{\{1, 2, 3, 4, 5\}\}$
- (c)  $\{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}\}$
- (d)  $\{\{1\}, \{2, 3\}, \{4\}\}$
- (e)  $\{\{\emptyset, 1, 2\}, \{3, 4, 5\}\}$

2.

Select all and only the correct statements about an equivalence relation  $E$  on a set  $U$ :

- (a)  $E \in U \times U$
- (b)  $E = U \times U$
- (c)  $E \subseteq U \times U$
- (d)  $\forall x \in U ([x]_E \in U)$
- (e)  $\forall x \in U ([x]_E \subseteq U)$
- (f)  $\forall x \in U ([x]_E \in \mathcal{P}(U))$
- (g)  $\forall x \in U ([x]_E \subseteq \mathcal{P}(U))$

# Wednesday December 1

**Scenario:** Good morning! You're a user experience engineer at Netflix. A product goal is to design customized home pages for groups of users who have similar interests. Your manager tasks you with designing an algorithm for producing a clustering of users based on their movie interests, so that customized homepages can be engineered for each group.

**Conventions for today:** We will use  $U = \{r_1, r_2, \dots, r_t\}$  to refer to an arbitrary set of user ratings (we'll pick some specific examples to explore) that are a subset of  $Rt_5$ . We will be interested in creating partitions  $C_1, \dots, C_m$  of  $U$ . We'll assume that each user represented by an element of  $U$  has a unique ratings tuple.

Your idea: equivalence relations! You offer your manager three great options:

$$E_{id} = \{ ( (x_1, x_2, x_3, x_4, x_5), (x_1, x_2, x_3, x_4, x_5) ) \mid (x_1, x_2, x_3, x_4, x_5) \in Rt_5 \}$$

*Describe how each homepage should be designed ...*

$$E_{proj} = \{ ( (x_1, x_2, x_3, x_4, x_5), (y_1, y_2, y_3, y_4, y_5) ) \in Rt_5 \times Rt_5 \mid (x_1 = y_1) \wedge (x_2 = y_2) \wedge (x_3 = y_3) \}$$

*Describe how each homepage should be designed ...*

$$E_{circ} = \{ (u, v) \in Rt_5 \times Rt_5 \mid d( ( (0, 0, 0, 0, 0), u ) ) = d( ( (0, 0, 0, 0, 0), v ) ) \}$$

*Describe how each homepage should be designed ...*

**Scenario:** Good morning! You're a user experience engineer at Netflix. A product goal is to design customized home pages for groups of users who have similar interests. You task your team with designing an algorithm for producing a clustering of users based on their movie interests. Your team implements two algorithms that produce different clusterings. How do you decide which one to use? What feedback do you give the team in order to help them improve? Clearly, you will need to use math.

Your idea: find a way to **score** clusterings (partitions)

**Definition:** For a cluster of ratings  $C = \{r_1, r_2, \dots, r_n\} \subseteq U$ , the **diameter** of the cluster is defined by:

$$\text{diameter}(C) = \max_{1 \leq i, j \leq n} (d(r_i, r_j))$$

Consider  $x = (1, 0, 1, 0, 1)$ ,  $y = (1, 1, 1, 0, 1)$ ,  $z = (-1, -1, 0, 0, 0)$ ,  $w = (0, 0, 0, 1, 0)$ .

What is  $\text{diameter}(\{x, y, z\})$ ?  $\text{diameter}(\{x, y\})$ ?  $\text{diameter}(\{x, z, w\})$ ?

*diameter* works on single clusters. One way to aggregate across a clustering  $C_1, \dots, C_m$  is  $\sum_{k=1}^m \text{diameter}(C_k)$

Is this a good score?

How can we express the idea of **many elements within a small area**? Key idea: "give credit" to small diameter clusters with many elements.

**Definition:** For a cluster of ratings  $C = \{r_1, r_2, \dots, r_n\} \subseteq U$ , the **density** of the cluster is defined by:

$$\frac{n}{1 + \text{diameter}(C)}$$

Can you use density to decide whether the partition given by the equivalence classes of  $E_{proj}$  or  $E_{circ}$  for this task?

# Looking forward

## Tips for future classes from the CSE 20 TAs and tutors

- In class
  - Go to class
  - Show up to class early because sometimes seats get taken/ the classroom gets full and then you have to sit on the floor
  - There's usually a space for skateboards/longboards/eboards to go at the front or rear of the lecture hall
  - If you have a flask water bottle please ensure that its secured during a lecture and it cannot fall - putting on the floor often leads to it falling since people sometimes cross your seats.
  - Take notes - it's much faster and more effective to note-take in class than watch recordings after, particularly if you do so longhand
  - Resist the urge to sit in the back. You will be able to focus much better sitting near the front, where there are fewer screens in front of you to distract from the lecture content
  - If you bring your laptop to class to take notes / access class materials, sit towards the back of the room to minimize distractions for people sitting behind you!
  - On zoom it's easy to just type a question out in chat, it might be a little more awkward to do so in person, but it is definitely worth it. Don't feel like you should already know what's being covered
  - Always check you have your iclicker<sup>1</sup> on you. Just keep it in your backpack permanently. That way you can never forget it.
  - Don't be afraid to talk to the people next to you during group discussions. Odds are they're as nervous as you are, and you can all benefit from sharing your thoughts and understanding of the material
  - Certain classes will podcast the lectures, just like Zoom archives lecture recordings, at [podcast.ucsd.edu](http://podcast.ucsd.edu)
  - If they aren't podcasted, and you want to record lectures, ask your professor for consent first
- Office hours, tutor hours, and the CSE building
  - Office hours are a good place to hang out and get work done while being able to ask questions as they come up
  - Office hour attendance is typically much busier in person (and confined to the space in the room)
  - Get to know the CSE building: CSE B260, basement labs, office hours rooms
  - Know how to get in to the building after-hours
- Libraries and on-campus resources
  - Look up what library floors are for what, how to book rooms: East wing of Geisel is open 24/7 (they might ask to see an ID if you stay late), East Wing of Geisel has chess boards and jigsaw puzzles, study pods on the 8th floor, free computers/wifi

---

<sup>1</sup>iclickers are used in many classes to encourage active participation in class. They're remotes that allow you to respond to multiple choice questions and the instructor can show a histogram of responses in real time.



- Know Biomed exists and is usually less crowded
- Most libraries allow you to borrow whiteboards and markers (also laptops, tablets, microphones, and other cool stuff) for 24 hours
- Take advantage of Dine with a prof / Coffee with a prof program. It's legit free food / coffee once per quarter.
- When planning out your daily schedule, think about where classes are, how much time will they take, are their places to eat nearby and how you can schedule social time with friends to nearby areas
- Take into account the distances between classes if they are back to back
- Final exams
  - What are 8am finals? Basically in-person exams are different
  - Don't forget your university card during exams
  - Blue books for exams (what they are, where to get them)
  - Seating assignments for exams and go early to make sure you're in the right place (check the exits to make sure you're reading it the right way)
  - Know where your exam is being held (find it on a map at least a day beforehand). Finals are often in strange places that take a while to find

## CSE department course numbering system

### Lower division

- CSE 12, Basic Data Structures and Object-Oriented Programming
- CSE 15L (2 units), Software Tools and Technique Laboratory
- CSE 20 or Math 15A, Introduction to Discrete Mathematics
- CSE 21 Mathematics for Algorithms and Systems
- CSE 30, Computer Organization & Systems Programming

### Upper division

- Advanced Data Structures and Programming: CSE 100
- Theory and Algorithms: CSE 101, CSE 105
- Software Engineering: CSE 110, CSE 112
- Systems/Networks: CSE 120 or CSE 123 or CSE 124
- Programming Languages /Databases: CSE 130 or CSE 132A
- Security/Cryptography: CSE 107 or CSE 127
- AI / Machine Learning/ Vision/ Graphics: CSE 150A or CSE 151A or CSE 151B or CSE 152A or CSE 158 or CSE 167
- Hardware / Architecture: CSE 140/ CSE 140L Components and Design Techniques for Digital Systems Architecture, CSE 141 / CSE 141L Introduction to Computer Architecture and CSE 141L Project in Computer Architecture (2 units), CSE 142 / CSE 142L Comp Arch Software Perspective

## Review

1.

Assume there are five movies in the database, so that each user's ratings can be represented as a 5-tuple. We call  $Rt_5$  the set of all ratings 5-tuples. Consider the binary relation on the set of all 5-tuples where each component of the 5-tuple is an element of the collection  $\{-1, 0, 1\}$

$$G_1 = \{(u, v) \mid \text{the ratings of users } u \text{ and } v \text{ agree about the first movie in the database}\}$$

$$G_2 = \{(u, v) \mid \text{the ratings of users } u \text{ and } v \text{ agree about at least two movies}\}$$

- (a) **True** or **False**: The relation  $G_1$  holds of  $u = (1, 1, 1, 1, 1)$  and  $v = (-1, -1, -1, -1, -1)$
- (b) **True** or **False**: The relation  $G_2$  holds of  $u = (1, 0, 1, 0, -1)$  and  $v = (-1, 0, 1, -1, -1)$
- (c) **True** or **False**:  $G_1$  is reflexive; namely,  $\forall u ( (u, u) \in G_1 )$
- (d) **True** or **False**:  $G_1$  is symmetric; namely,  $\forall u \forall v ( (u, v) \in G_1 \rightarrow (v, u) \in G_1 )$
- (e) **True** or **False**:  $G_1$  is transitive; namely,  $\forall u \forall v \forall w ( ((u, v) \in G_1 \wedge (v, w) \in G_1) \rightarrow (u, w) \in G_1 )$
- (f) **True** or **False**:  $G_2$  is reflexive; namely,  $\forall u ( (u, u) \in G_2 )$
- (g) **True** or **False**:  $G_2$  is symmetric; namely,  $\forall u \forall v ( (u, v) \in G_2 \rightarrow (v, u) \in G_2 )$
- (h) **True** or **False**:  $G_2$  is transitive; namely,  $\forall u \forall v \forall w ( ((u, v) \in G_2 \wedge (v, w) \in G_2) \rightarrow (u, w) \in G_2 )$

2.

Recall that in a movie recommendation system, each user's ratings of movies is represented as a  $n$ -tuple (with the positive integer  $n$  being the number of movies in the database), and each component of the  $n$ -tuple is an element of the collection  $\{-1, 0, 1\}$ .

We call  $Rt_5$  the set of all ratings 5-tuples.

Define  $d : Rt_5 \times Rt_5 \rightarrow \mathbb{N}$  by

$$d( ( (x_1, x_2, x_3, x_4, x_5), (y_1, y_2, y_3, y_4, y_5) ) ) = \sum_{i=1}^5 |x_i - y_i|$$

For a cluster of ratings  $C = \{r_1, r_2, \dots, r_n\} \subseteq U$ , the **diameter** of the cluster is defined by:

$$diameter(C) = \max_{1 \leq i, j \leq n} (d( (r_i, r_j) ))$$

- (a) Calculate  $diameter(\{(-1, -1, -1, -1, 1), (0, 0, 0, 0, 0), (1, 1, 1, 1, 1)\})$
- (b) What's the greatest (possible) diameter of a collection that has exactly two (distinct) ratings 5-tuples?
- (c) What's the least (possible) diameter of a collection that has exactly two (distinct) ratings 5-tuples?

## Friday December 3

Convert  $(2A)_{16}$  to

- binary (base \_\_\_\_)
- decimal (base \_\_\_\_)
- octal (base \_\_\_\_)
- ternary (base \_\_\_\_)

The bases of RNA strands are elements of the set  $B = \{\mathbf{A}, \mathbf{C}, \mathbf{G}, \mathbf{U}\}$ . 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.

Each of the sets below is described using set builder notation. Rewrite them using the roster method.

- $\{s \in S \mid \text{the leftmost base in } s \text{ is the same as the rightmost base in } s \text{ and } s \text{ has length } 3\}$
- $\{s \in S \mid \text{there are twice as many As as Cs in } s \text{ and } s \text{ has length } 1\}$

Certain sequences of bases serve important biological functions in translating RNA to proteins. The following recursive definition gives a special set of RNA strands: The set of RNA strands  $\hat{S}$  is defined (recursively) by

Basis step:  $\mathbf{AUG} \in \hat{S}$   
Recursive step: If  $s \in \hat{S}$  and  $x \in R$ , then  $sx \in \hat{S}$

where  $R = \{\mathbf{UUU}, \mathbf{CUC}, \mathbf{AUC}, \mathbf{AUG}, \mathbf{GUU}, \mathbf{CCU}, \mathbf{GCU}, \mathbf{UGG}, \mathbf{GGA}\}$ .

Each of the sets below is described using set builder notation. Rewrite them using the roster method.

- $\{s \in \hat{S} \mid s \text{ has length less than or equal to } 5\}$
- $\{s \in S \mid \text{there are twice as many Cs as As in } s \text{ and } s \text{ has length } 6\}$

Let  $W = \mathcal{P}(\{1, 2, 3, 4, 5\})$ . Consider the statement

$$\forall A \in W \forall B \in W \forall C \in W ((A \cap B = A \cap C) \rightarrow (B = C))$$

Translate the statement to English. Negate the statement and translate this negation to English. Decide whether the original statement or its negation is true and justify your decision.

The set of linked lists of natural numbers  $L$  is defined by

$$\begin{array}{ll} \text{Basis step:} & [] \in L \\ \text{Recursive step:} & \text{If } l \in L \text{ and } n \in \mathbb{N}, \text{ then } (n, l) \in L \end{array}$$

The function  $length : L \rightarrow \mathbb{N}$  that computes the length of a list is

$$\begin{array}{ll} \text{Basis step:} & length([]) = 0 \\ \text{Recursive step:} & \text{If } l \in L \text{ and } n \in \mathbb{N}, \text{ then } length((n, l)) = 1 + length(l) \end{array}$$

Prove or disprove: the function  $length$  is onto.

Prove or disprove: the function  $length$  is one-to-one.

Suppose  $A$  and  $B$  are sets and  $A \subseteq B$ :

**True or False?** If  $A$  is infinite then  $B$  is finite.

**True or False?** If  $A$  is countable then  $B$  is countable.

**True or False?** If  $B$  is infinite then  $A$  is finite.

**True or False?** If  $B$  is uncountable then  $A$  is countable.

Compute the last digit of

$$(42)^{2021}$$

*Extra* Describe the pattern that helps you perform this computation and prove it using mathematical induction.



## Review

1. Please complete the CAPE and TA evaluations. Once you have done so complete the custom feedback form for this quarter: <https://forms.gle/pbYWSRDP2znkciM46>  
Then, (we're using the honor system here), write out the statement "I have completed the end of quarter evaluations" and you'll receive credit for this question.