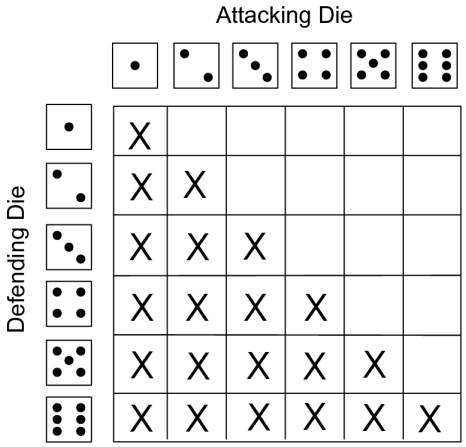
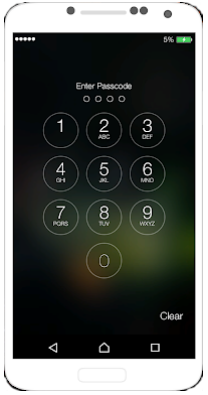




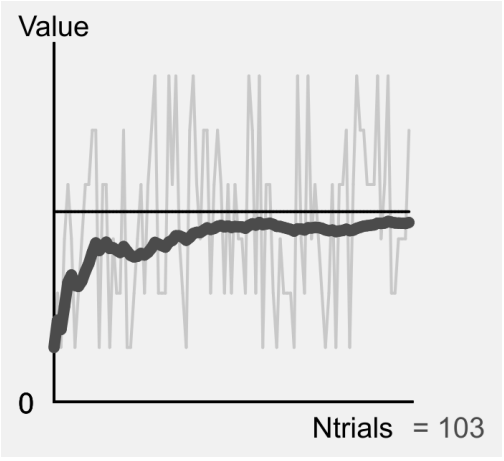
Crack the PIN

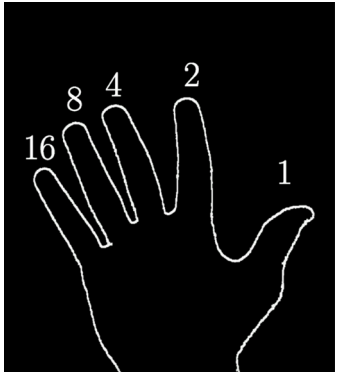
guess pin: 610

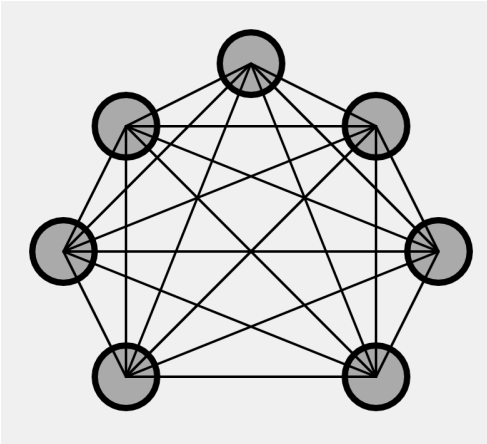
A match!

Progress bar (match may occur before scan is complete):

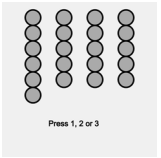
A Computer-Science Enriched Curriculum for Discrete Math



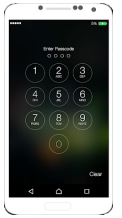




1. Games



2. Counting/Combinatorics
   - With application to cybersecurity



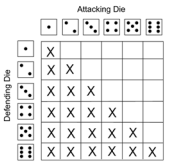
Crack the PIN

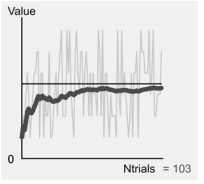
guess pin: 610

A match!

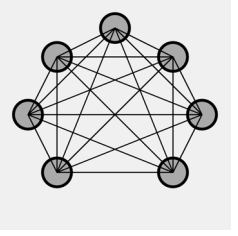
Progress bar (match may occur before scan is complete):

3. Probability





4. Connectivity

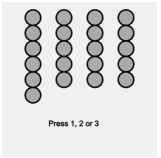


5. Iteration and recursion

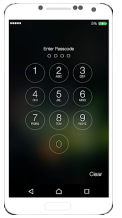
$$N! = 1 \cdot 2 \cdot 3 \cdots (N-1) \cdot N$$

6. Cryptography

1. **Games**



2. Counting/Combinatorics
   - With application to cybersecurity



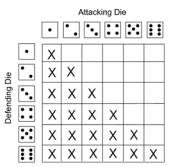
Crack the PIN

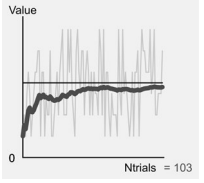
guess pin: 610

A match!

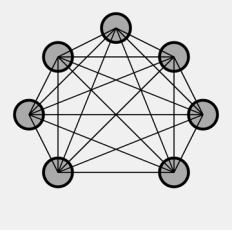
Progress bar (match may occur before scan is complete):

3. Probability





4. Connectivity
   - Related to Graph/Network Theory



5. Iteration and recursion

$$N! = 1 \cdot 2 \cdot 3 \cdots (N-1) \cdot N$$

6. Cryptography

Example Activity: Thai 21

21 Flags

Each team takes 1-3 flags

Winner takes the last flag

Classroom

- Start hands on playing the game (~2 days)
- Alternate sessions:
  - Coding (screens up)
  - Debrief of coding, strategy of game, discussion (screens down)

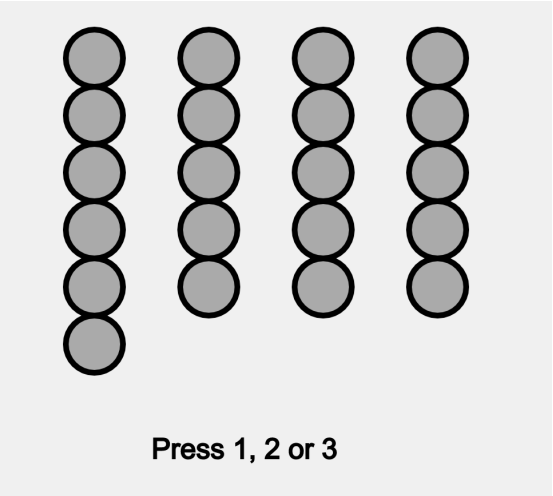
Note:
Screens down content modified from San Diego State Discrete Math Project Collaborative

Thai 21 Coding

Part 1 – Put the rules of the game in the code

Part 2 – Keep track of turns

Part 3 – Player vs Computer (AI)



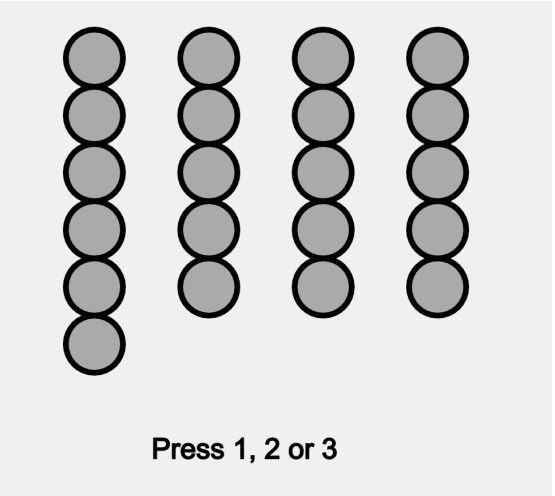
Total class time Parts 1-3:
~Three weeks

Thai 21 Coding

Part 1 – Put the rules of the game in the code

Part 2 – Keep track of turns

Part 3 – Player vs Computer (AI)



Link to activity:

http://go.osu.edu/thai21

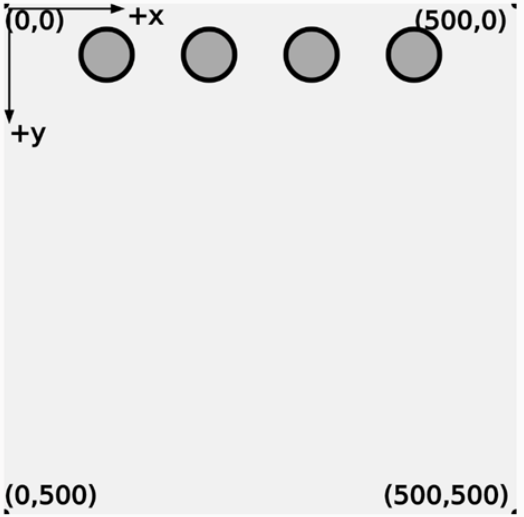
Browser-based JavaScript Coding Interface

Auto-refresh Thai 21 v0 part1 milestones by ChrisOrban

sketch.js

```
1  showaxes = true;
2
3  function draw(){
4
5    clearscreen(240,240,240); // light gray background
6    display();
7
8    // add code below to remove flags by pressing 1,2,3
9
10   // Set the color of the flags
11   fill(170,170,170); // dark gray
12   stroke(0,0,0); // black edge
13
14   // Define the flags below
15   drawFlag(100,50); // x,y
16   drawFlag(200,50);
17   drawFlag(300,50);
18   drawFlag(400,50);
19
20
21 } // end draw() DO NOT ADD ANY CODE AFTER THIS LINE!!!
22
23
```

Preview



OBJECTIVE: Draw one flag **STATUS:** ✅

OBJECTIVE: Draw more than one flag **STATUS:** ✅

OBJECTIVE: Draw more than four flags **STATUS:** ❌

OBJECTIVE: Draw 21 flags **STATUS:** ❌

OBJECTIVE: Add a way to remove 1, 2 or 3 flags **STATUS:**

Compatible with:

mac OS iPadOS chromebook Linux

Requires physical keyboard!

Auto-refresh Thai 21 v0 part1 milestones by ChrisOrban

sketch.js

```
showaxes = false;

function draw(){

  clearscreen(240,240,240); // light gray background
  display();

  // add code below to remove flags by pressing 1,2,3

  // Set the color of the flags
  fill(170,170,170); // dark gray
  stroke(0,0,0); // black edge

  // Define the flags below
  drawFlag(100,50); // x,y

} // end draw() DO NOT ADD ANY CODE AFTER THIS LINE!!!
```

Preview

OBJECTIVE: Draw one flag **STATUS:** ✅

OBJECTIVE: Draw more than one flag **STATUS:** ❌

OBJECTIVE: Draw more than four flags **STATUS:** ❌

OBJECTIVE: Draw 21 flags **STATUS:** ❌

OBJECTIVE: Add a way to remove 1, 2 or 3 flags **STATUS:**

Here is some code you can add to plot four flags in a row:

```
y = 50;
for (x = 100 ; x <= 400; x += 100) {
 drawFlag(x,y);
}
```

Copy code to clipboard

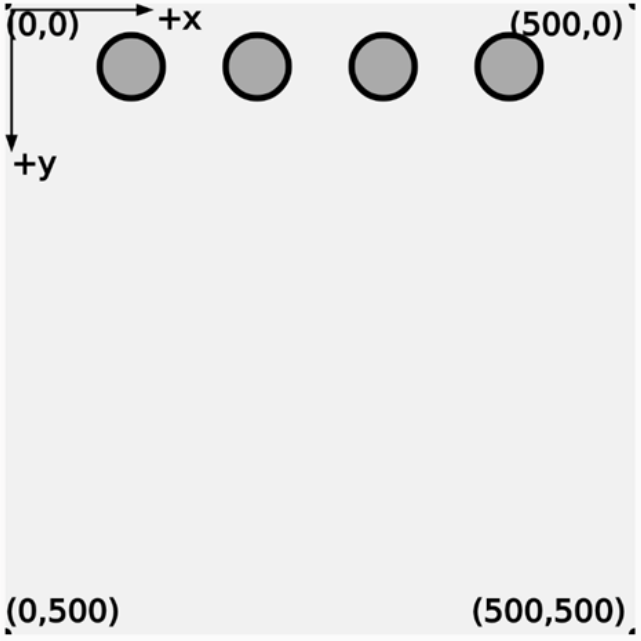
Auto-refresh Thai 21 v0 part1 milestones by ChrisOrban

sketch.js

```
1  showaxes = true;
2
3  function draw(){
4
5    clearscreen(240,240,240); // light gray background
6    display();
7
8    // add code below to remove flags by pressing 1,2,3
9
10   // Set the color of the flags
11   fill(170,170,170); // dark gray
12   stroke(0,0,0); // black edge
13
14   // Define the flags below
15   y = 50;
16   for (x = 100 ; x <= 400; x += 100) {
17    drawFlag(x,y);
18   }
19
20
21 } // end draw() DO NOT ADD ANY CODE AFTER THIS LINE!!!
22
23
```

For loop!

Preview



OBJECTIVE: Draw one flag **STATUS:** ✅

OBJECTIVE: Draw more than one flag **STATUS:** ✅

OBJECTIVE: Draw more than four flags **STATUS:** ❌

OBJECTIVE: Draw 21 flags **STATUS:** ❌

OBJECTIVE: Add a way to remove 1, 2 or 3 flags **STATUS:**

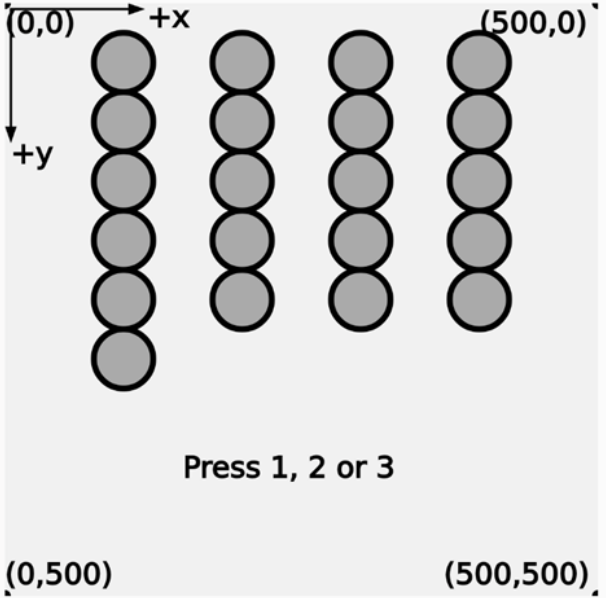
Fast Forward

Auto-refresh Thai 21 v0 part1 milestones by ChrisOrban

sketch.js

```
showaxes = true;
Ntotalflags = 21;

function draw(){

  clearscreen(240,240,240); // light gray background
  display();

  // add code below to remove flags by pressing 1,2,3
  drawText('Press 1, 2 or 3',150,400);
  if ((keyIsPressed == true) && (key == 1)) Ntotalflags -= 1;
  if ((keyIsPressed == true) && (key == 2)) Ntotalflags -= 2;
  if ((keyIsPressed == true) && (key == 3)) Ntotalflags -= 3;

  // Set the color of the flags
  fill(170,170,170); // dark gray
  stroke(0,0,0); // black edge

  // Define the flags below
   Nflags_so_far = 0;
  for (y = 50; y <= 400; y += 50) {
   for (x = 100 ; x <= 400; x += 100) {
    if ( Nflags_so_far < Ntotalflags ) {
     drawFlag(x,y);
     Nflags_so_far += 1;
    }
   }
  }

} // end draw() DO NOT ADD ANY CODE AFTER THIS LINE!!!
```

Preview



OBJECTIVE: Draw one flag **STATUS:** ✅

OBJECTIVE: Draw more than one flag **STATUS:** ✅

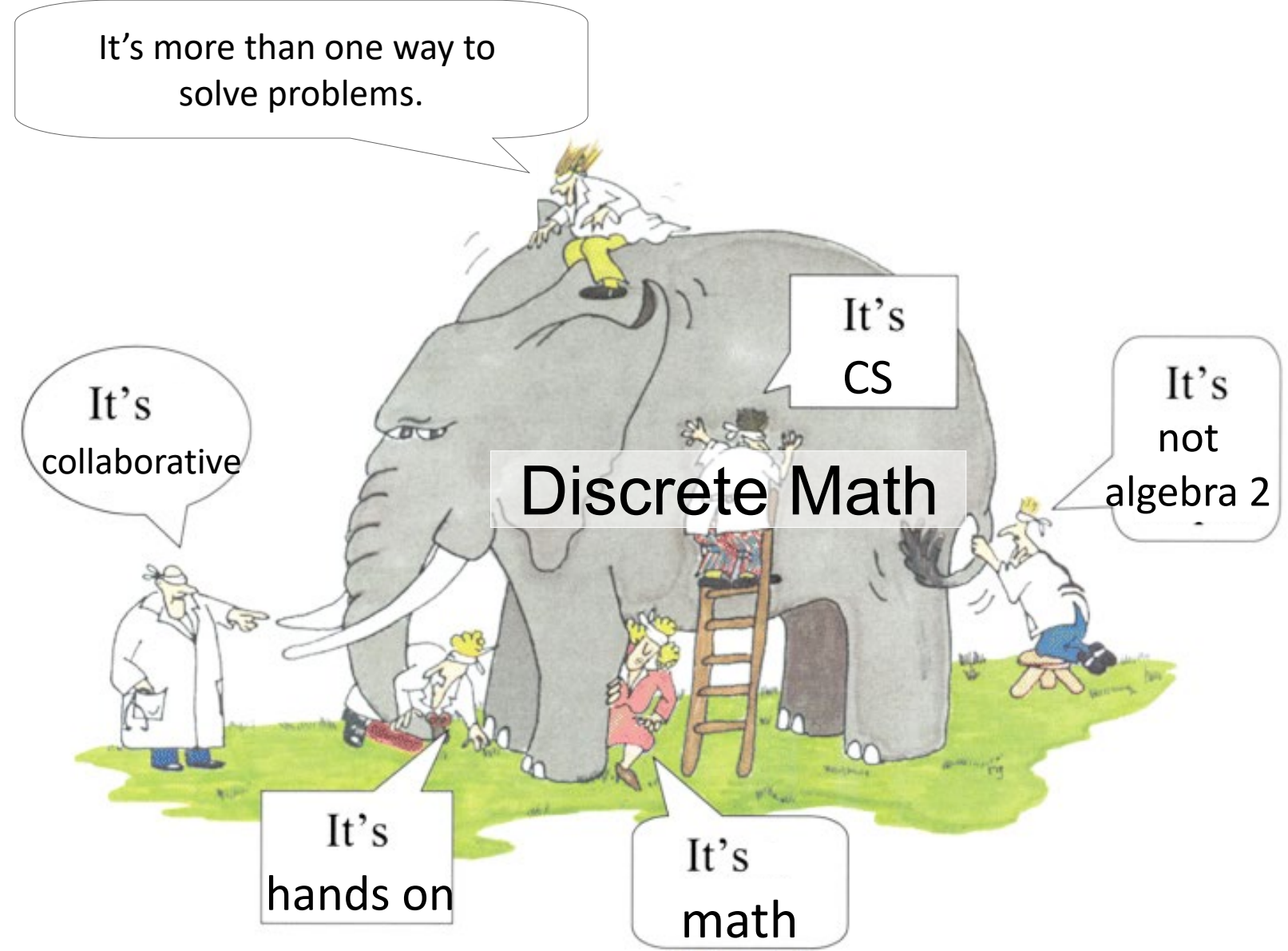
OBJECTIVE: Draw more than four flags **STATUS:** ✅

OBJECTIVE: Draw 21 flags **STATUS:** ✅

OBJECTIVE: Add a way to remove 1, 2 or 3 flags **STATUS:**

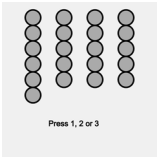
Where is the math?

- Coordinate system
- Transformation of coordinates in drawFlag(x,y);
- Boolean logic of if statements
  - Encoding the rules of the game
- Visualizing the strategy with rows of 4
- Later we will write code so a computer opponent follows the strategy
- The strategy involves calculating the reminder of remaining flags/4

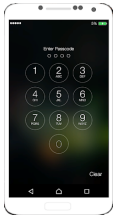

It’s more than one way to solve problems.
It’s CS
It’s collaborative
Discrete Math
It’s not algebra 2
It’s hands on
It’s math

Year Long Curriculum

1. Games



2. Counting/Combinatorics
   - With application to cybersecurity



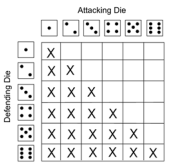
Crack the PIN

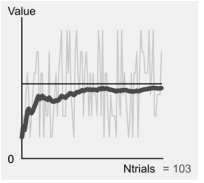
guess pin: 610

A match!

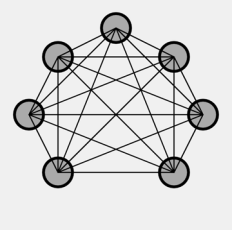
Progress bar (match may occur before scan is complete):

3. Probability





4. Connectivity
   - Related to Graph/Network Theory



5. Iteration and recursion

$$N! = 1 \cdot 2 \cdot 3 \cdots (N-1) \cdot N$$

6. Cryptography

Questions?

Comments?

Feel free to email me at orban.14@osu.edu

Link to activity: http://go.osu.edu/thai21

Extra Slides

UNIVERSITY OF CALIFORNIA SAN DIEGO

CALIFORNIA STATE UNIVERSITY SAN MARCOS

Math is No Longer a Four-Letter Word: A Mixed Methods Study of Two Non-Traditional Fourth-Year Mathematics Classes

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Education

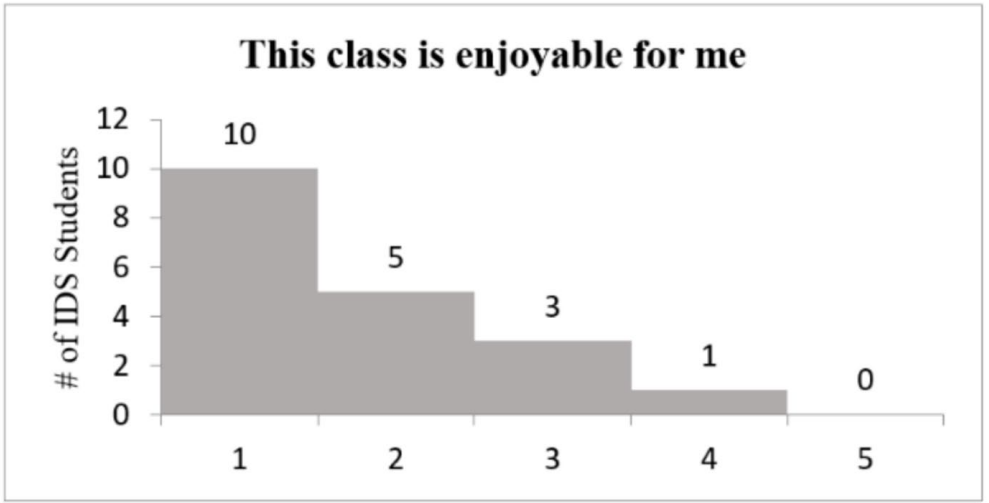
in

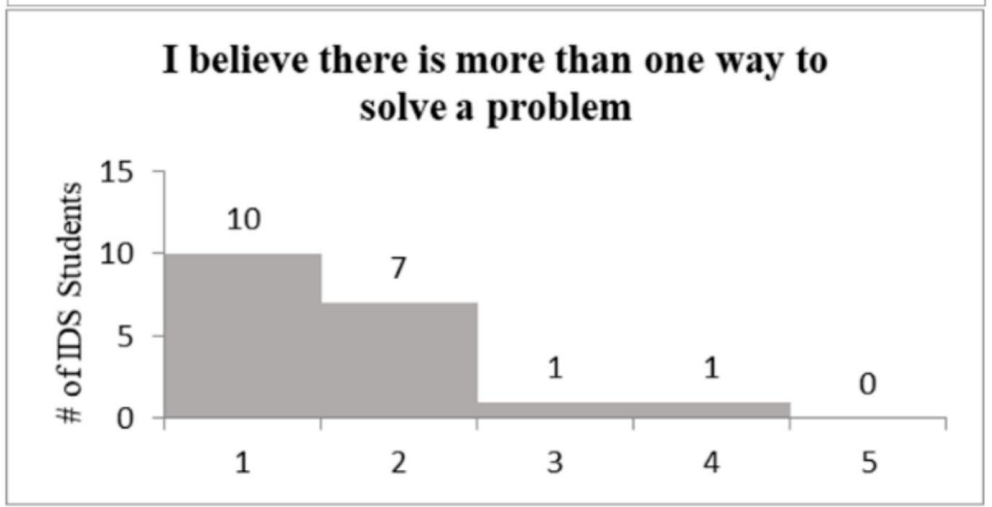
Educational Leadership

by

Erica Heinzman

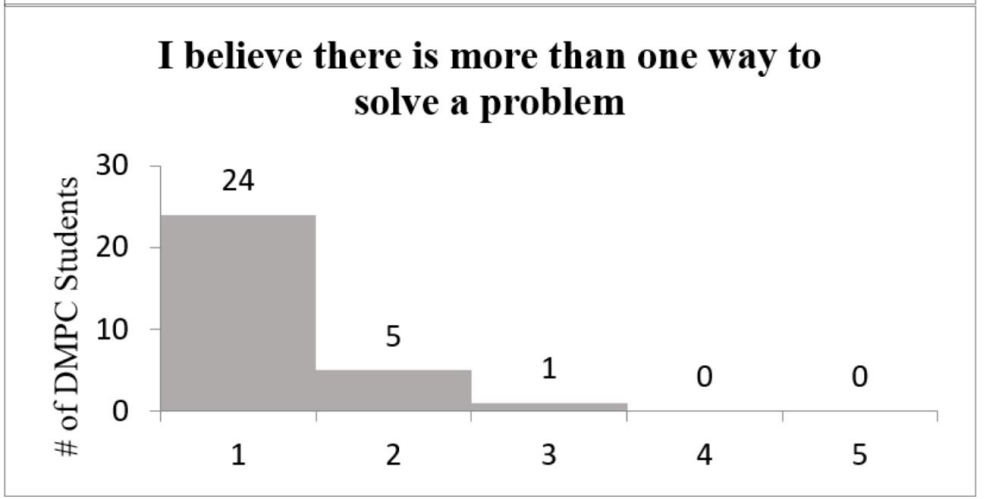
Data Science



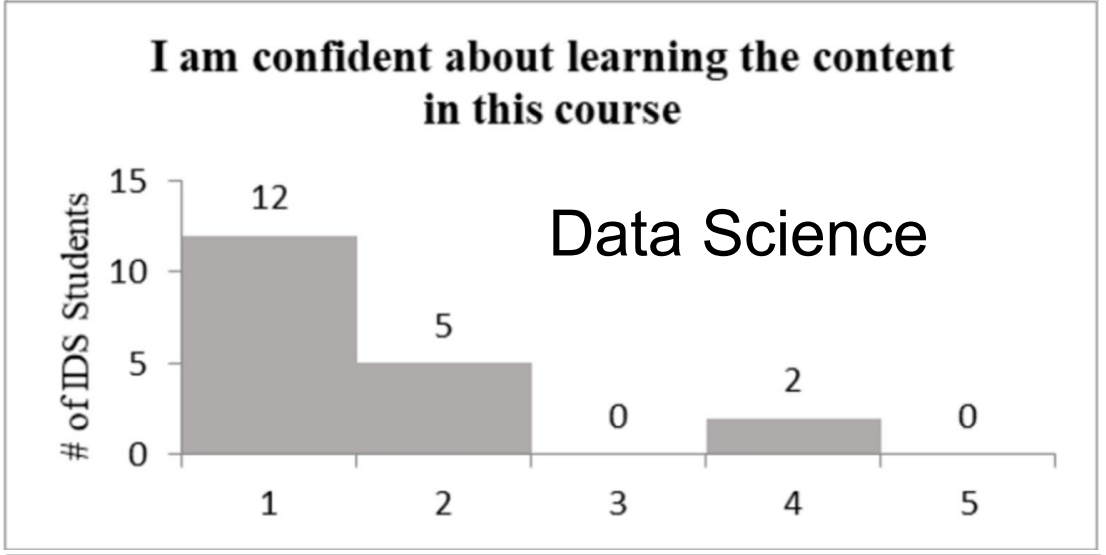


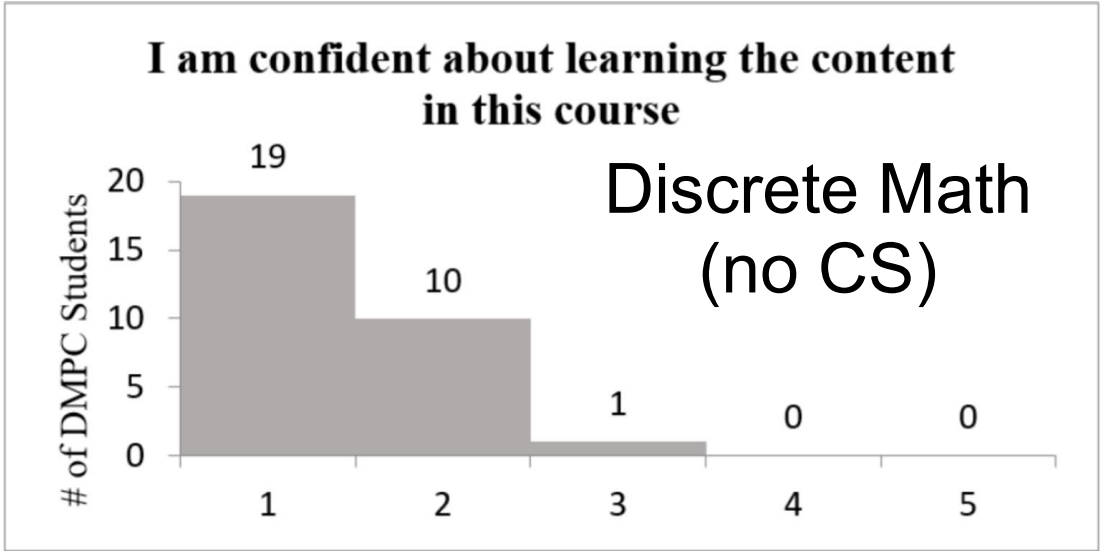
1 = “strongly agree”

Discrete Math (no CS)



5 = “strongly disagree”





1 = “strongly agree” 5 = “strongly disagree”

Brief Summary

- Data Science and Discrete math are both efforts to change what it feels like to be in a math class
- This is in much the same spirit as the quantitative reasoning effort in Ohio
- Students work with each other to solve problems