Iteration 1

Problem Identification

We need to be able to plan and prepare every year for the "seasonal" flu, and be ready for the next "pandemic." How can computational thinking help us make the best use of limited resources?

To set up your identified problem

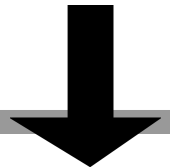
Decomposition (How would you break down your problem into sub-problems?

Subproblem:
- Can we predict the number of people who will get infected (or have complications) each year?

Pattern Recognition (Are there related solutions to draw on?)

Abstraction (How would you abstract this problem?)

Graphic Organizer



Iteration 2

Problem Identification

Can we build a model to predict the number of people that will be infected with the flu?

To set up your identified problem

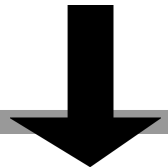
Decomposition (How would you break down your problem into sub-problems?)

Pattern Recognition (Are there related solutions to draw on?)

Abstraction (How would you abstract this problem?)

Information that's important to our problem:
- Number of people susceptible to the flu
- Number of people that are infected
- Number of people that have been vaccinated
- Rate of transmission
- Rate of recovery

Graphic Organizer



Iteration 3

Problem Identification

Based on the SIR model, how do we create an algorithm that will predict the number of people who will be infected?

To set up your identified problem

Decomposition (How would you break down your problem into sub-problems?)

Pattern Recognition (Are there related solutions to draw on?)

Two patterns that can help solve our problem:

1. People that are susceptible (S) can become Infected (I). This occurs ate rate b.

2. Infected people (I) can become Recovered (R). This occurs at rate r.

Abstraction (How would you abstract this problem?)

Graphic Organizer

Iteration 4

Problem Identification

To set up your identified problem

Decomposition (How would you break down your problem into sub-problems?)

Pattern Recognition (Are there related solutions to draw on?)

Abstraction (How would you abstract this problem?)

Graphic Organizer

Iteration 5

Problem Identification

To set up your identified problem

Decomposition (How would you break down your problem into sub-problems?)

Pattern Recognition (Are there related solutions to draw on?)

Abstraction (How would you abstract this problem?)

Graphic Organizer

Iteration 6

Problem Identification

To set up your identified problem

Decomposition (How would you break down your problem into sub-problems?)

Pattern Recognition (Are there related solutions to draw on?)

Abstraction (How would you abstract this problem?)

Graphic Organizer

Iteration 7

Problem Identification

To set up your identified problem

Decomposition (How would you break down your problem into sub-problems?)

Pattern Recognition (Are there related solutions to draw on?)

Abstraction (How would you abstract this problem?)

Graphic Organizer

Iteration 8

Problem Identification

To set up your identified problem

Decomposition (How would you break down your problem into sub-problems?)

Pattern Recognition (Are there related solutions to draw on?)

Abstraction (How would you abstract this problem?)

Graphic Organizer

Iteration 9

Problem Identification

To set up your identified problem

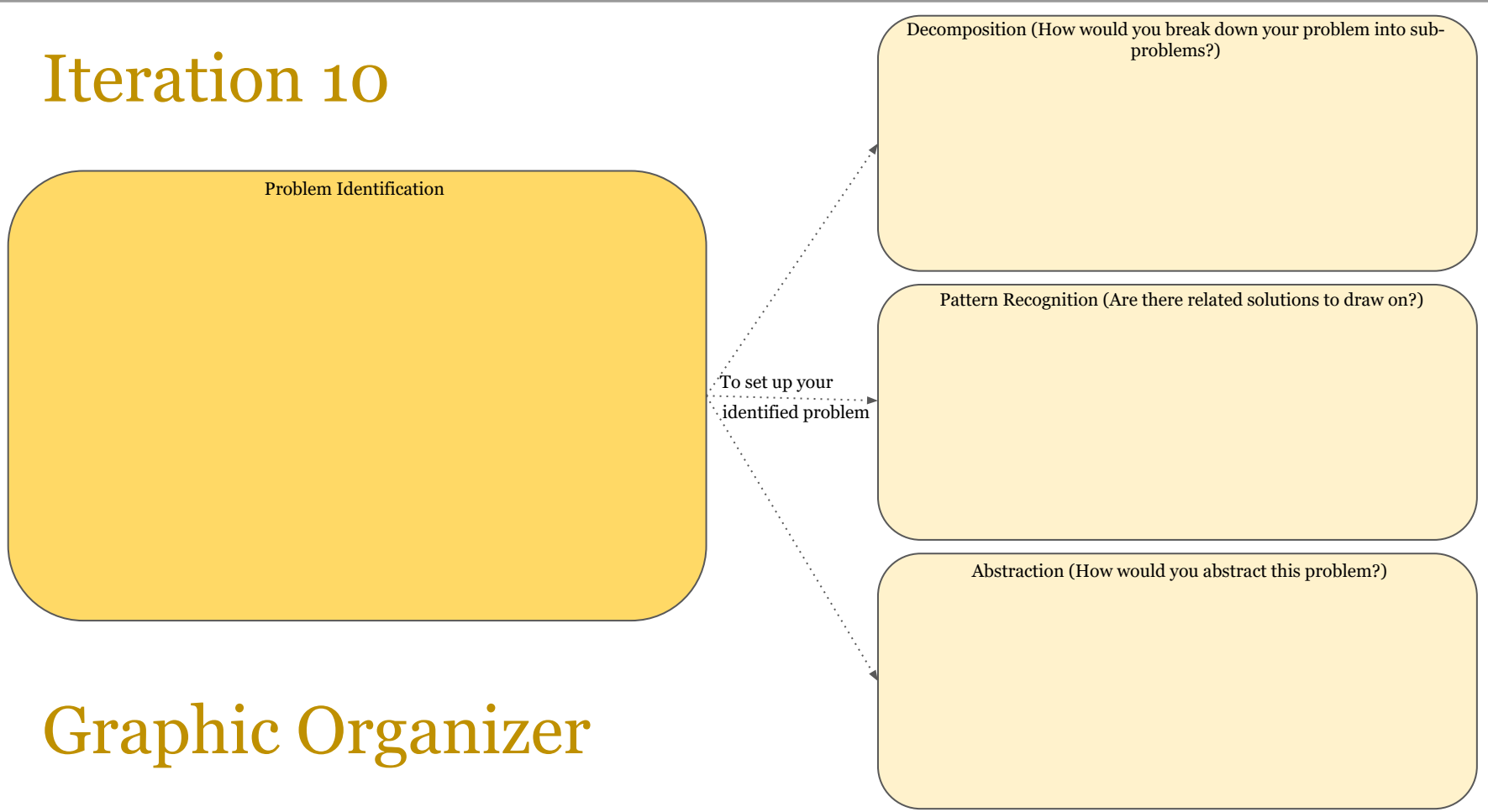
Decomposition (How would you break down your problem into sub-problems?)

Pattern Recognition (Are there related solutions to draw on?)

Abstraction (How would you abstract this problem?)

Graphic Organizer

Iteration 10



Graphic Organizer