Codebreaker Lesson Plan

Summary

The game Mastermind (figure 1) is a plastic puzzle game in which one player (the codemaker) comes up with a secret code consisting of 4 colors chosen from red, green, blue, yellow, black, and white (repeats allowed, order matters). The other player (the codebreaker) tries to guess the code in as few tries as possible. This lesson uses a similar game, which we call Codebreaker, that can be used in the classroom to explore combinatorics and computational thinking.

Figure 1: Mastermind

Time Line

What	**Time Required**	**Where**
Introduction	10 minutes	Lesson plan
Codebreaker Activity	20 minutes	CodebreakerWorksheet.pdf
Discussion with optional demo	15 minutes	CodebreakerSlides.pdf http://csunplugged.mines.edu/codebreaker/game.cgi
What's it all about	5 minutes	Lesson plan
Total time	50 minutes	

Materials Required

- ✓ Pen/pencil
- ✓ Codebreaker instructions are optional (page 1. If printed, one per pair of students.)
- ✓ Secret Code Slips (Page 2. Print single-sided, one copy per each pair of students.)
- ✓ Codebreaker Worksheet (Page 3. Each student needs one; possibly print double-sided because only one student is codebreaker at a time)
- ✓ Computer & projection system (optional, for Codebreaker computer demo)

Purpose

This activity explores the concept of *algorithm*, a precise set of steps to accomplish a task, and *data representation*, how to effectively represent information to assist in solving a problem.

Cheat Sheet

Terminology

- *Combinatorics* - a branch of mathematics dealing with combinations of objects. Combinational problems arise in many fields, including computer science.
- *Data representation* - representing data in a way that makes it easy to use, such as to quickly find patterns or make decisions.
- *Algorithm* - step by step process for solving a problem.

Introduction - Whole Class

Lesson Vocabulary (you may want to write on the board)

- Algorithm

Start by asking if anyone has played the game Mastermind. If yes, see what the students remember. You may want to write the main points for the Codebreaker game (almost exactly the same as Mastermind) on the board, or just discuss:

1. One person is the codemaker (this is Person A on the worksheet)
    - In the board game, this person comes up with a secret code consisting of 4 colors chosen from red, green, blue, yellow, black, and white.
    - Colors may be repeated, and order matters.
    - For our game, we will represent codes as 4 digits, where each digit has a value from 1 to 6 (because there are 6 colors, e.g., 1 could be red, 2 blue, etc.).
2. The second person is the codebreaker (Person B), who tries to guess the code in as few guesses as possible.
3. During game play, the codemaker determines a secret code and writes it on the Secret Code Slip (to avoid forgetting or changing the code).
4. The codebreaker then "submits" a guess to the codemaker.
5. The codemaker provides feedback in the form of 2 numbers (also illustrated in figure 2):
    - The first number tells how many digits are the CORRECT VALUE and in the CORRECT LOCATION.
    - The second number tells how many digits are the CORRECT VALUE but in an INCORRECT LOCATION
6. If the response is 4-0 (all numbers correct and in correct location), codebreaker wins!
7. If this is the 10^{th} guess, codebreaker loses.
8. Repeat from step 4.

Codebreaker Demo

Now demonstrate the game with the students. Write a secret code on a piece of paper. Then ask students to guess your secret code. You may want to continue until the students guess your code, OR stop as soon as you think all students understand how the game is played.

Figure 2 shows a sample demonstration of the game (just to show how you would write this on the board; it's unlikely the students would make these exact guesses).

The secret code for this demonstration is **1321**.

Guess	Response	Explanation
1234	1-2	1 is right location, 2 and 3 are wrong location, 4 not in code
1254	1-1	1 is right location, 2 is wrong location, 4 and 5 not in code
1123	2-2	1 and 2 right location, 1 and 3 wrong location
1321	4-0	This is it!

Figure 2: Sample Codebreaker game

Activity - Codebreaker

Students should now play Codebreaker. This can be done in pairs, or you may have a group of 3, if there are an odd number of students. Each pair of students should have one **Secret Code** page and two copies of the **Codebreaker's worksheet** (can be on two sides of same page). This allows each person to be codemaker and codebreaker.

You may also want to give each pair of students a copy of the Codebreaker Instructions (generally this should not be needed if the students understand the demo).

Discussion - Whole Class

After students have participated in the activity, have them discuss the questions below. Questions and answers are included on the associated slides, which you may use as you see fit.

1. Suppose you play as the codemaker, what kind of code would you choose? Why?

   This question is fairly open ended. From a mathematical standpoint, all codes are equally as hard to guess for an intelligent player, but knowing your opponent's strategies can lead to some codes that may be harder for your opponent to guess.

   In tests against our computer player, codes with 3 color repeats (e.g., 2111) were harder to guess, requiring an average of about 4.7 turns to guess, whereas most codes were guessed in an average of 4.4 turns. This may not be true when two humans are playing.

2. How many possible codes are there?

   Depending on the mathematical background of the students, this may be a review of a topic they already know, or a brief glimpse into combinatorics. There are five slides related to this question, to help walk students through the reasoning.

   Answer: $6^4 = 1296$

   How to arrive at this? There are 4 numbers in the code, and each can be chosen from 1 to 6. So for the first number there are 6 choices. Since numbers can be repeated, there are also 6 choices for the second digit. So for 2 digits there would be 6*6 or 36 possibilities. Expanding this to four digits yields 6^4 possible combinations.

   This calculation would change if different sets of digits are valid. For example, if the code contained only digits 1,2,3,4 then there would be $4^4 = 256$ codes.

3. Suppose you play as the codebreaker, what's an effective strategy? What's a good first guess?

   This question is less open ended than the first one (how to choose a good code if you are codemaker). It really asks which codes will reduce the expected number of turns the game will take after the first move. An investigation of the entire game tree of codebreaker shows that selecting a code with two each of two distinct digits (e.g., 1122 or 1331) will not only reduce the set of remaining codes the most on average, but will lower the expected number of moves when compared to other starting plays.

   We can think roughly about the information that would be provided in response to such a guess, compared to a guess which, for example, included four unique digits. Since there are only 2 digits, the possible responses will inform the guesser that either

   A. neither digit is included. So the two digits from the first guess should never be used again, and the second guess could be two other digits.

   B. at least one digit is correct

   From this point there are various strategies that may be used to explore (i.e., there is not one "correct" process for human guessers, but they should follow some process of deduction/elimination). There are nine slides that walk through this type of strategy.

4. Are there strategies for computer programs that are more effective than human player strategies?

    Computer Player demo (optional)

    Come up with a secret code and run the computer program:

    http://csunplugged.mines.edu/codebreaker/game.cgi

    Notice how the included computer program guesses the code. You may want to repeat this a few times.

    Yes, plenty of computer programs exist that can codebreak and win in a very small amount of moves (including the one included with this activity). A simple algorithm (step-by-step process) exists for Codebreaker (actually developed for the Mastermind game). Let *S* be the set of all possible codes:

    1. Select a random code from *S* and ask the codemaker for response.
    2. If the codemaker gives a winning response (i.e,. 4-0), we have won. Stop.
    3. Remove all elements from *S* which would no longer be possible under the codemaker's response.
    4. Goto step 2

    For example, suppose that the secret code was 1321, the computer had played a few guesses, and now *S* = {1222, 1322, 1221, 1231}. Then we guess 1222 randomly from *S*, and get a response of 2-0. We remove 1222 from S since we just guessed it, remove 1322 from S since if the secret code was 1322 and we guessed 1222, we would receive a response of 3-0, which was not what we received, and, for the same reason, remove 1221. This leaves us with *S* = {1231}, so on our next guess we will win.

    So how well does this strategy preform? Figure 3 shows a frequency histogram of a computer player playing against this strategy with many random codes. As you can see from the histogram, the code is guessed in five tries or less about 85% of the time.

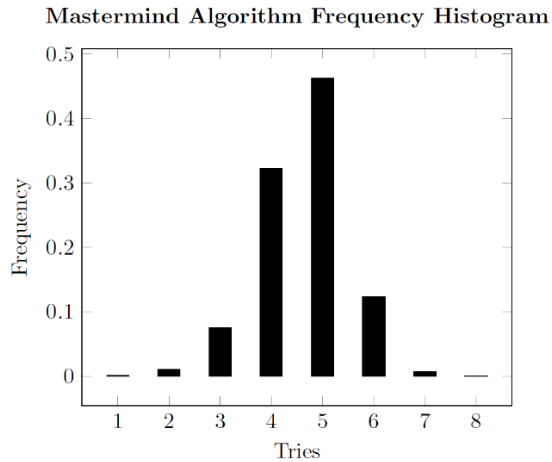

Figure 3

You may have already improvised a way to improve this strategy: rather than choosing randomly from *S*, what if we chose with an "educated guess"? In this case, we would select randomly from the set of code(s) which, when compared to all the other codes, have the lowest average cardinality of the remaining elements in *S* for all possible responses. In other words, we simply attempt to narrow the set of remaining codes more quickly. When this is done, the computer player wins in five tries or less about 97% of the time, as seen in Figure 4.

This algorithm falls into the category of *exhaustive search*, which is practical for computer programs but not for human players (it's unlikely that any human would write down 1296 codes, then manually eliminate them).

Computer programs have been developed that are guaranteed to solve the Mastermind problem in at most 5 guesses (and can sometimes break the code in fewer guesses).

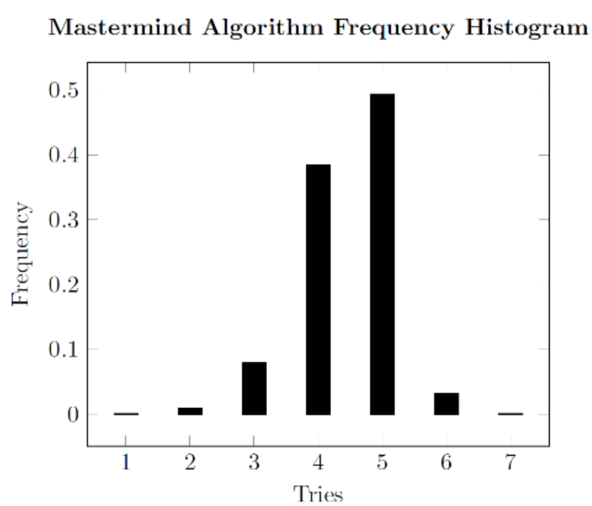

Figure 4

What's It All About Discussion - Whole Class

The human strategy we showed for Codebreaker is similar to *troubleshooting*. When something is not working (e.g., a computer program, a printer, a mechanical device), you might be tempted to try multiple things all at once to try to fix it. A better strategy is to change just one thing at a time, so you can determine the effect.

The codes in Mastermind are kind of like passwords. We have passwords to safeguard lots of personal information.

- Would you want everyone to know your grades?
- Would your parents want others to be able to access their bank accounts?

Assume your local bank decides that for simplicity their passwords will be just 4 digits. How secure is that?

- What does it mean to be secure?
- How easy would it be for a human to crack this password?
- How easy would it be for a computer to crack this password?
- What actions have companies taken to guard against this type of brute-force effort?
- What types of information do people sometimes encode in their passwords, that make it easier for others to guess?

Possible links to more background material:

- https://en.wikipedia.org/wiki/Enigma_machine
- https://www.coursera.org/learn/principles-of-computing-1/lecture/ge171/combinatorics-and-password-breaking