Power Round

November 19, 2017

The Rules of Mastermind

MasterMind is a two-player code-breaking game. One of the players is the *code-setter*, and the other player acts as the *code-breaker*.

At the beginning of the game, the code-setter generates a secret code kept hidden from the code-breaker. The code consists of 4 pegs, each of which can be one of the six colors red, blue, green, yellow, orange, or violet. Throughout this round, these possible colors will be indicated by R, B, G, Y, O and V respectively. The code is allowed to contain repeated colors.

For example, RRBB, GYVO, RRRV, BYBO, and GGGG are all valid codes.

On the other hand, RRBK and GGGYB are examples of invalid codes that cannot be used in the game. The first code RRBK is not valid because it uses a label "K" not in our color list. The second code GGGYB is not a valid code because it consists of five colored pegs instead of four colored pegs.

The code-breaker will attempt to guess the secret code in a series of guess and feedback turns. Each turn proceeds as follows:

(i) The code-breaker guesses a code.

(ii) The code-maker will give a feedback (c, m) on the guess by telling the code-breaker the number c of pegs in the guess that had the *correct* color and position, and the number m of pegs in the guess that were correctly colored but *misplaced*.

These turns are repeated until the code-breaker correctly guesses the identity of secret-code. The objective of the code-breaker is to figure out the secret-code using as few guesses as possible.

For reference, below is a sample game, played out in its entirety, with the guesses and feedbacks presented for each turn.

Secret Code: RBYV

Turn	Guess	Correct	Misplaced
1	RRBB	1	1
2	GYRB	0	3
3	BRGO	0	2
4	RBVY	2	2
5	RBYV	4	0

In the above game, the code-setter picks RBYV as the secret code, and the code-breaker begins by guessing RRBB.

We observe that for this first guess, the first R peg is in the correct position. Moreover, the B peg is a color used in the secret code, but in the guess it is misplaced (it should go in the second slot instead of the third or fourth slot). So, the feedback that the code-breaker receives for this first guess is the ordered pair $(1,1)$, corresponding to the information (**Correct**: 1, **Misplaced**: 1).

Note that even though there were 2 B pegs from guess the in the wrong position, the feedback was not $(1,2)$. This is because only one of the B pegs could potentially be moved to fill in the second slot and match with the secret code. So, one of the B pegs in the guess was "misplaced", and the other was just incorrect - neither correct or misplaced.

Note also that when giving the feedback, the code-setter does **not** tell the code-breaker which pegs are correct and which are misplaced!

1 The Codes of MasterMind (7 pts)

1.1. How many possible secret codes can be generated using 4 pegs and 6 colors? **(2 pts)**

1.2. Observe that for any possible feedback (c,m), we must have $c+m \leq 4$. How many ordered pairs of nonnegative integers (c,m) correspond to a valid feedback that might be given in the game of MasterMind? **(3 pts)**

1.3. As it turns out, there is exactly one ordered pair of nonnegative integers (c,m) with $c+m=4$ that is not attainable as a valid feedback in the game. What is this ordered pair and why is it not a valid feedback that could be received? **(2 pts)**

2 The Combinatorics of MasterMind (25 pts)

When playing the game, it can be helpful for the code-breaker to think about the set of possible secret codes that are consistent with all the feedback that they have received in the previous rounds. At the start of the game, before any guesses are made, the secret code could be any valid code. However, as the code-breaker makes more guesses, the number of possible secret codes diminishes.

For example, if the first guess is RRBB, and the feedback is $(1,1)$, then the code-breaker now knows that RRRB is not the secret code. This is because if RRRB were the secret code, then the feedback for the guess RRBB would have been $(3,0)$ instead.

2.1. If the first guess is GGGG, and the feedback is (0, 0), how many secret codes are still possible? **(2 pts)**

2.2. If the first guess is GGGY, and the feedback is (1, 0), how many secret codes are still possible? **(3 pts)**

2.3. If the first guess is GGYY, and the feedback is (1, 0), how many secret codes are still possible? **(3 pts)**

2.4. Each of the following game boards shows the first few rounds of a different game of MasterMind. In each case find, with justification, how many secret codes are still possible for that game.

(a) **(5 pts)**

Turn	Guess	Correct	Misplaced
1	RRRR	0	0
2	BBBB	2	0
3	GGGG	0	0
4	YYYY	0	0
5	OOOO	1	0

(b) **(5 pts)**

Turn	Guess	Correct	Misplaced
1	RRRR	1	0
2	BBBB	1	0
3	YYYY	1	0
4	GGGG	1	0
5	ROOO	0	1
6	BOOO	0	1
7	GOOO	0	1

(c) **(7 pts)**

Turn	Guess	Correct	Misplaced
1	RRBB	2	2
2	RBBR	2	2

3 The Strategy of MasterMind (45 pts)

In 1976, mathematician and computer scientist Donald Knuth found a strategy which allows the code-breaker in MasterMind to always correctly guess the secret code in 5 or fewer turns. The questions in this section will work towards proving the weaker result that there is a deterministic strategy for the code-breaker that can always guess the secret code in 12 or fewer turns.

One subproblem of guessing the secret code is to determine the color composition of the secret code. That is, the code-breaker would like to know for each color how many pegs of that color there are: how many R pegs are in the secret code, how many B pegs are in the secret code, etc.

3.1. Give a simple, deterministic (non-random) strategy that the codebreaker can use to always determine the color composition of the secret code in 6 or fewer guesses. **(2 pts)**

3.2. Give a deterministic strategy that can always determine the color composition of the secret code in 5 or fewer guesses. **(2 pts)**

Suppose now that you are the code-breaker. Your friend was able to peek at the secret code, and they tell you that all 4 pegs in the code have the same color. Unfortunately, your friend forgot which color the pegs were. Luckily for you however, there are only 6 possibilities for the secret code: RRRR, OOOO, YYYY, GGGG, BBBB, or VVVV. So, as the code breaker, you could trivially guess the code in 6 or fewer guesses just by guessing all 6 of these possible codes in some order.

3.3. Give a deterministic strategy that you can use as the codebreaker to always guess the secret code in 5 or fewer guesses, given this information (hint: you do not have to guess all 4 pegs of the same color on the first turn!). **(5 pts)**

3.4. Improve the strategy from problem 3.3 to get a strategy that you can use as the codebreaker to always guess the secret code in 3 or fewer guesses, given this information **(8 pts)**

We move on to considering strategies for the code-breaker in the general MasterMind game, where we aren't guaranteed that all 4 pegs are of the same color. In coming up with strategies in the general case, it might help to use the ideas from the previous problems about color composition.

3.5. For the standard Mastermind game, give a deterministic strategy for the codebreaker that can always guess the secret code in strictly less than 1300 turns. **(3 pts)**

3.6. Give a deterministic strategy for the codebreaker that can always guess the secret code in 30 or fewer guesses (hint: $30 = 6 + 4!$, there are 6 possible colors and 4 pegs in the code). **(10 pts)**

3.7. Improve the strategy from problem 3.6 to get a strategy that guarantees the code-breaker can always guess the secret code in 20 or fewer guesses. **(7 pts)**

3.8. Improve the strategy from problem 3.6 to get a strategy that guarantees the code-breaker can always guess the secret code in 12 or fewer guesses. **(8 pts)**

4 The Bounds of MasterMind (13 pts)

In this final section we will prove that there is no strategy for the code-breaker that guarantees that they can always guess the secret code in 3 or fewer guesses.

At the beginning of the game, there are N possible secret codes the code-setter could set (where N is the answer to problem 1.1), and there are M possible feedbacks (where M is the answer to problem 1.2). After the code-breaker makes the first guess, but before they receive the first feedback, we can partition the set of N possible codes into M sets, based on which feedback the code-breaker would see if a particular code was the secret code.

4.1. Show that no matter what the first guess is, there exists some feedback such that there will still be at least 80 possible secret codes after that feedback is received. **(3 pts)**

4.2. Using the result of problem 4.1, show that there cannot be a deterministic strategy that always guesses the secret code in 2 or fewer guesses. **(3 pts)**

4.3. Extend the reasoning from problems 4.1 and 4.2 to show that there cannot be a deterministic strategy that always guesses the secret code in 3 or fewer guesses. **(7 pts)**