MASTERMIND REVISITED

David Kauchak
CS52 – Spring 2017

Admin

Assignment 7

CS52 mixer

Mastermind revisited

Leverage some of these ideas for Mastermind!

3 colors, 3 pegs

3 Colors: Red, Green, Blue

3 pegs: [____, ____, ____]

How many different codes?

3 colors, 3 pegs

$27!$ (colors^{pegs} = 3^3)

[Red, Red, Red]	[Green, Red, Red]	[Blue, Red, Red]
[Red, Red, Green]	[Green, Red, Green]	[Blue, Red, Green]
[Red, Red, Blue]	[Green, Red, Blue]	[Blue, Red, Blue]
[Red, Green, Red]	[Green, Green, Red]	[Blue, Green, Red]
[Red, Green, Green]	[Green, Green, Green]	[Blue, Green, Green]
[Red, Green, Blue]	[Green, Green, Blue]	[Blue, Green, Blue]
[Red, Blue, Red]	[Green, Blue, Red]	[Blue, Blue, Red]
[Red, Blue, Green]	[Green, Blue, Green]	[Blue, Blue, Green]
[Red, Blue, Blue]	[Green, Blue, Blue]	[Blue, Blue, Blue]

3 colors, 3 pegs

[Red, Red, Red]	[Green, Red, Red]	[Blue, Red, Red]
[Red, Red, Green]	[Green, Red, Green]	[Blue, Red, Green]
[Red, Red, Blue]	[Green, Red, Blue]	[Blue, Red, Blue]
[Red, Green, Red]	[Green, Green, Red]	[Blue, Green, Red]
[Red, Green, Green]	[Green, Green, Green]	[Blue, Green, Green]
[Red, Green, Blue]	[Green, Green, Blue]	[Blue, Green, Blue]
[Red, Blue, Red]	[Green, Blue, Red]	[Blue, Blue, Red]
[Red, Blue, Green]	[Green, Blue, Green]	[Blue, Blue, Green]
[Red, Blue, Blue]	[Green, Blue, Blue]	[Blue, Blue, Blue]

Codemaker chooses this code

Naïve approach (assignment 3)

What would our naïve approach guess first?

[Red, Red, Red]	[Green, Red, Red]	[Blue, Red, Red]
[Red, Red, Green]	[Green, Red, Green]	[Blue, Red, Green]
[Red, Red, Blue]	[Green, Red, Blue]	[Blue, Red, Blue]
[Red, Green, Red]	[Green, Green, Red]	[Blue, Green, Red]
[Red, Green, Green]	[Green, Green, Green]	[Blue, Green, Green]
[Red, Green, Blue]	[Green, Green, Blue]	[Blue, Green, Blue]
[Red, Blue, Red]	[Green, Blue, Red]	[Blue, Blue, Red]
[Red, Blue, Green]	[Green, Blue, Green]	[Blue, Blue, Green]
[Red, Blue, Blue]	[Green, Blue, Blue]	[Blue, Blue, Blue]

Naïve approach (assignment 3)

Guess 1: [Red, Red, Red]

Response? (codemaker)

Exact Inexact

[Red, Red, Red]	[Green, Red, Red]	[Blue, Red, Red]
[Red, Red, Green]	[Green, Red, Green]	[Blue, Red, Green]
[Red, Red, Blue]	[Green, Red, Blue]	[Blue, Red, Blue]
[Red, Green, Red]	[Green, Green, Red]	[Blue, Green, Red]
[Red, Green, Green]	[Green, Green, Green]	[Blue, Green, Green]
[Red, Green, Blue]	[Green, Green, Blue]	[Blue, Green, Blue]
[Red, Blue, Red]	[Green, Blue, Red]	[Blue, Blue, Red]
[Red, Blue, Green]	[Green, Blue, Green]	[Blue, Blue, Green]
[Red, Blue, Blue]	[Green, Blue, Blue]	[Blue, Blue, Blue]

Naïve approach (assignment 3)

Guess 1: [Red, Red, Red] Response (codemaker) — Exact: 0, Inexact: 0

[Red, Red, Red]	[Green, Red, Red]	[Blue, Red, Red]
[Red, Red, Green]	[Green, Red, Green]	[Blue, Red, Green]
[Red, Red, Blue]	[Green, Red, Blue]	[Blue, Red, Blue]
[Red, Green, Red]	[Green, Green, Red]	[Blue, Green, Red]
[Red, Green, Green]	[Green, Green, Green]	[Blue, Green, Green]
[Red, Green, Blue]	[Green, Green, Blue]	[Blue, Green, Blue]
[Red, Blue, Red]	[Green, Blue, Red]	[Blue, Blue, Red]
[Red, Blue, Green]	[Green, Blue, Green]	[Blue, Blue, Green]
[Red, Blue, Blue]	[Green, Blue, Blue]	[Blue, Blue, Blue]

Which ones can we eliminate?

Naïve approach (assignment 3)

Guess 1: [Red, Red, Red] Response (codemaker) — Exact: 0, Inexact: 0

[Red, Red, Red]	[Green, Red, Red]	[Blue, Red, Red]
[Red, Red, Green]	[Green, Red, Green]	[Blue, Red, Green]
[Red, Red, Blue]	[Green, Red, Blue]	[Blue, Red, Blue]
[Red, Green, Red]	[Green, Green, Red]	[Blue, Green, Red]
[Red, Green, Green]	[Green, Green, Green]	[Blue, Green, Green]
[Red, Green, Blue]	[Green, Green, Blue]	[Blue, Green, Blue]
[Red, Blue, Red]	[Green, Blue, Red]	[Blue, Blue, Red]
[Red, Blue, Green]	[Green, Blue, Green]	[Blue, Blue, Green]
[Red, Blue, Blue]	[Green, Blue, Blue]	[Blue, Blue, Blue]

Any with red in them: 19 removed

Naïve approach (assignment 3)

What would our naïve approach guess next?

[Green, Green, Green]	[Blue, Green, Green]
[Green, Green, Blue]	[Blue, Green, Blue]
[Green, Blue, Green]	[Blue, Blue, Green]
[Green, Blue, Blue]	[Blue, Blue, Blue]

Naïve approach (assignment 3)

Guess 2: [Green, Green, Green] Response? (codemaker) — Exact, Inexact

[Green, Green, Green]	[Blue, Green, Green]
[Green, Green, Blue]	[Blue, Green, Blue]
[Green, Blue, Green]	[Blue, Blue, Green]
[Green, Blue, Blue]	[Blue, Blue, Blue]

Naïve approach (assignment 3)

Exact Inexact

Guess 2: [Green, Green, Green] Response? (codemaker) 1 0

[Green, Green, Green] [Blue, Green, Green]
[Green, Green, Blue] [Blue, Green, Blue]

[Green, Blue, Green] [Blue, Blue, Green]
[Green, Blue, Blue] [Blue, Blue, Blue]

Which ones can we eliminate?

Naïve approach (assignment 3)

Exact Inexact

Guess 2: [Green, Green, Green] Response? (codemaker) 1 0

[Green, Green, Green] [Blue, Green, Green]
[Green, Green, Blue] [Blue, Green, Blue]

[Green, Blue, Green] [Blue, Blue, Green]
[Green, Blue, Blue] [Blue, Blue, Blue]

Must have one green: removed 5.

Naïve approach (assignment 3)

What would our naïve approach guess next?

[Blue, Green, Blue]

[Blue, Blue, Green]

[Green, Blue, Blue]

Naïve approach (assignment 3)

Exact Inexact

Guess 3: [Green, Blue, Blue] Response? (codemaker)

[Blue, Green, Blue]

[Blue, Blue, Green]

[Green, Blue, Blue]

Naïve approach (assignment 3)

Guess 3: [Green, Blue, Blue]

Response? (codemaker)

Exact 1 Inexact 2

[Blue, Green, Blue]

[Blue, Blue, Green]

[Green, Blue, Blue]

Which ones can we eliminate?

Naïve approach (assignment 3)

Guess 3: [Green, Blue, Blue]

Response? (codemaker)

Exact 1 Inexact 2

[Blue, Green, Blue]

[Blue, Blue, Green]

[Green, Blue, Blue]

Only 1!

Naïve approach (assignment 3)

What would our naïve approach guess next?

[Blue, Green, Blue]

[Blue, Blue, Green]

Naïve approach (assignment 3)

Guess 4: [Blue, Green, Blue]

Response? (codemaker)

Exact Inexact

[Blue, Green, Blue]

[Blue, Blue, Green]

Naïve approach (assignment 3)

		Exact	Inexact
Guess 4: [Blue, Green, Blue]	Response? (codemaker)	1	2

[Blue, Green, Blue]

[Blue, Blue, Green]

Naïve approach (assignment 3)

		Exact	Inexact
Guess 5: [Blue, Blue, Green]	Response? (codemaker)	3	0

[Blue, Blue, Green]

Naïve approach (assignment 3)

It took us 5 guesses.

Guess 1: [Red, Red, Red]
Guess 2: [Green, Green, Green]
Guess 3: [Green, Blue, Blue]
Guess4: [Blue, Green, Blue]
Guess 5: [Blue, Blue, Green]

Can we do better (less guesses)?

Mastermind as adversarial search

We're the codebreaker (i.e. the person trying to guess the code)

Guess 1: [Red, Red, Red]
Guess 2: [Green, Green, Green]
Guess 3: [Green, Blue, Blue]
Guess4: [Blue, Green, Blue]
Guess 5: [Blue, Blue, Green]

What are codes that we *can* guess to get information?
What codes does the naïve algorithm pick from?

Mastermind as adversarial search

We're the codebreaker (i.e. the person trying to guess the code)

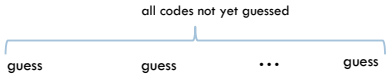
Guess 1: [Red, Red, Red]
Guess 2: [Green, Green, Green]
Guess 3: [Green, Blue, Blue]
Guess4: [Blue, Green, Blue]
Guess 5: [Blue, Blue, Green]

We can guess ***any*** code that we haven't previously guessed!
- For our last guess, we must guess the code
- For the other guesses, our goal is to gather information

The naïve algorithm only picks from codes that *could be the solution*

Mastermind as adversarial search

On our turn we could guess *any* code not already guessed



Challenge: we don't know what response we will get for a given guess

But we know we will get *a response*

Mastermind: a better approach

Response? (codemaker)	Exact	Inexact

What are all the possible responses to a guess (for 3 colors, 3 pegs)?

Mastermind: a better approach

Response? (codemaker)	Exact	Inexact
	0	0
	0	1
	0	2
	0	3
	1	0
	1	1
	1	2
	2	0
	3	0

Mastermind as adversarial search

On our turn we could guess *any* code not already guessed

Looking closer at a guess

What information does the codemaker response give us?

Looking closer at a guess

Helps us prune the possible valid codes

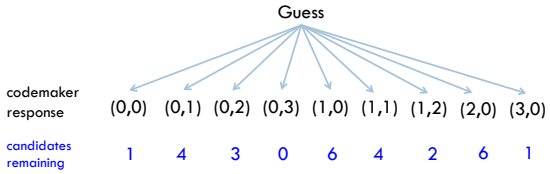
Naïve approach (assignment 3)

Guess 2: [Green, Green, Green]	Response? (codemaker)	Exact 1	Inexact 0
[Green, Green, Green]			[Blue, Green, Green]
[Green, Green, Blue]			[Blue, Green, Blue]
[Green, Blue, Green]			[Blue, Blue, Green]
[Green, Blue, Blue]			[Blue, Blue, Blue]

Must have one green: removed 5.

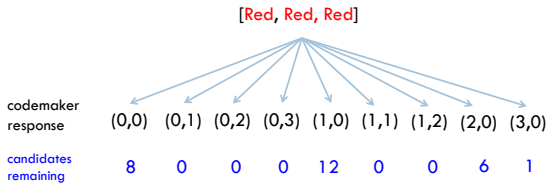
For a given code and response, we can calculate how many candidates are removed (e.g. 5 above) or, conversely, how many would still remain (e.g. 3 above)

Looking closer at a guess

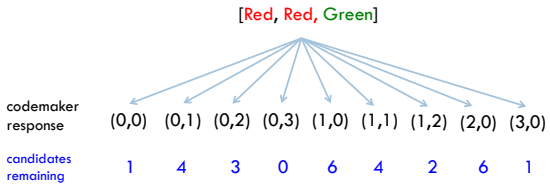


For a given guess, we can then calculate how many candidates would remain *if we got that response from the codemaker* for every possible response code

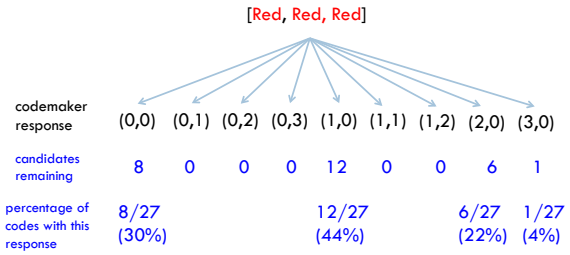
Which is better: option 1



Which is better: option 2

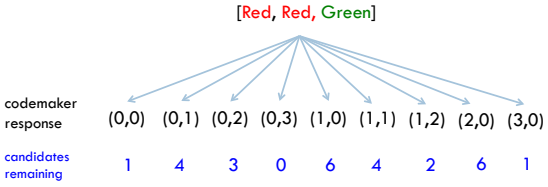


Naïve approach



Can view this as a distribution over possible codes
Ideally, we'd like it to be as evenly distributed as possible.

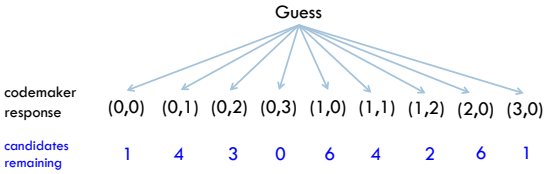
Improved approach



Much more evenly distributed

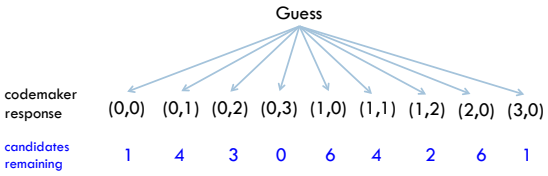
Also, in the worst case, only 6 remaining

Quantifying distribution



How can we quantify the "quality" of the guess?

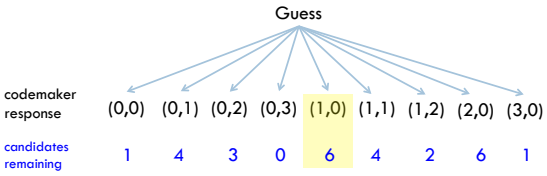
Looking closer at a guess



One heuristic uses the minimax idea: view codemaker as the "opponent"

What is the worst case scenario if we make this guess?

Looking closer at a guess

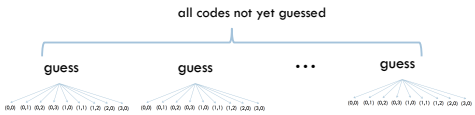


Minimax idea: view codemaker as the "opponent"

Worse case, the response is (1,0) and there are 6 remaining.

Mastermind as adversarial search

On our turn we could guess *any* code not already guessed



Which guess should we make?

Mastermind as adversarial search

On our turn we could guess *any* code not already guessed

The one that minimizes the maximum remaining candidates

Max (codemaker response): assume we get the response with the largest remaining candidate set

Min (our guess): pick the one that, worst case, results in the smallest candidate set

Improved approach

Guess 1: [Red, Red, Green] Response? (codemaker) Exact Inexact

[Red, Red, Red]	[Green, Red, Red]	[Blue, Red, Red]
[Red, Red, Green]	[Green, Red, Green]	[Blue, Red, Green]
[Red, Red, Blue]	[Green, Red, Blue]	[Blue, Red, Blue]
[Red, Green, Red]	[Green, Green, Red]	[Blue, Green, Red]
[Red, Green, Green]	[Green, Green, Green]	[Blue, Green, Green]
[Red, Green, Blue]	[Green, Green, Blue]	[Blue, Green, Blue]
[Red, Blue, Red]	[Green, Blue, Red]	[Blue, Blue, Red]
[Red, Blue, Green]	[Green, Blue, Green]	[Blue, Blue, Green]
[Red, Blue, Blue]	[Green, Blue, Blue]	[Blue, Blue, Blue]

Improved approach

Guess 1: [Red, Red, Green] Response? (codemaker) Exact 1 Inexact 0

[Red, Red, Red]	[Green, Red, Red]	[Blue, Red, Red]
[Red, Red, Green]	[Green, Red, Green]	[Blue, Red, Green]
[Red, Red, Blue]	[Green, Red, Blue]	[Blue, Red, Blue]
[Red, Green, Red]	[Green, Green, Red]	[Blue, Green, Red]
[Red, Green, Green]	[Green, Green, Green]	[Blue, Green, Green]
[Red, Green, Blue]	[Green, Green, Blue]	[Blue, Green, Blue]
[Red, Blue, Red]	[Green, Blue, Red]	[Blue, Blue, Red]
[Red, Blue, Green]	[Green, Blue, Green]	[Blue, Blue, Green]
[Red, Blue, Blue]	[Green, Blue, Blue]	[Blue, Blue, Blue]

Which ones can we eliminate?

Improved approach

Guess 1: [Red, Red, Green] Response? (codemaker) Exact 1 Inexact 0

[Red, Red, Red]	[Green, Red, Red]	[Blue, Red, Red]
[Red, Red, Green]	[Green, Red, Green]	[Blue, Red, Green]
[Red, Red, Blue]	[Green, Red, Blue]	[Blue, Red, Blue]
[Red, Green, Red]	[Green, Green, Red]	[Blue, Green, Red]
[Red, Green, Green]	[Green, Green, Green]	[Blue, Green, Green]
[Red, Green, Blue]	[Green, Green, Blue]	[Blue, Green, Blue]
[Red, Blue, Red]	[Green, Blue, Red]	[Blue, Blue, Red]
[Red, Blue, Green]	[Green, Blue, Green]	[Blue, Blue, Green]
[Red, Blue, Blue]	[Green, Blue, Blue]	[Blue, Blue, Blue]

Filtered out 21: only 6 codes remain!

Improved approach

Guess 2: [Red, Blue, Blue] Response? (codemaker) Exact 1 Inexact 1

		[Blue, Red, Blue]
	[Green, Green, Green]	[Blue, Green, Green]
	[Green, Blue, Green]	[Blue, Blue, Green]
[Red, Blue, Blue]		

Which ones can we eliminate?

Improved approach

Guess 2: [Red, Blue, Blue] Response? (codemaker) Exact 1 Inexact 1

		[Blue, Red, Blue]
	[Green, Green, Green]	[Blue, Green, Green]
	[Green, Blue, Green]	[Blue, Blue, Green]
[Red, Blue, Blue]		

All of them except the answer are filtered

Improved approach (3 colors, 3 pegs)

Naïve approach	Improved approach
Guess 1: [Red, Red, Red]	Guess 1: [Red, Red, Green]
Guess 2: [Green, Green, Green]	Guess 2: [Red, Blue, Blue]
Guess 3: [Green, Blue, Blue]	Guess 3: [Blue, Blue, Green]
Guess 4: [Blue, Green, Blue]	
Guess 5: [Blue, Blue, Green]	
5 guesses	3 guesses
Guaranteed: at most 5 On average: 3.30	Guaranteed: at most 4 On average: 2.74

Improved approach (6 colors, 4 pegs)

For 6 colors and 4 pegs:

Naïve approach

- Worst case: 9 guesses
- On average: 5.765 guesses

Improved approach

- Worst case: 5 guesses
- On average: 4.476

Improved approach

Published by Donald Knuth in 1977

https://sakai.claremont.edu/access/content/group/CX_mtg_94136/resources/knuth-mastermind.pdf

Key heuristic

The one that minimizes the maximum remaining candidates

Max (codemaker response): assume we get the response with the largest remaining candidate set

Min (our guess): pick the one that, worst case, results in the smallest candidate set

How do we calculate this?

Key heuristic

The one that minimizes the maximum remaining candidates

min
- For all codes not yet guessed:
  - max
    - Consider all possible responses:
      - Calculate the size of the remaining candidates if we guessed that code and got that response
    - select response with largest remaining for that code
- select code with smallest max

Key heuristic

min
- For all codes not yet guessed:
  - max
    - Consider all possible responses:
      - Calculate the size of the remaining candidates if we guessed that code and got that response
    - select response with largest remaining for that code
- select code with smallest max

All possible codes. Only reduces by 1 each time.

All possible *valid* codes given the responses so far.

Why do we consider codes that are not possible candidates?

Key heuristic

min
- For all codes not yet guessed:
  - max
    - Consider all possible responses:
      - Calculate the size of the remaining candidates if we guessed that code and got that response
    - select response with largest remaining for that code
- select code with smallest max

All possible codes. Only reduces by 1 each time.

All possible *valid* codes given the responses so far.

Even though we know they're not correct, they might still be helpful in reducing the set of possible candidates

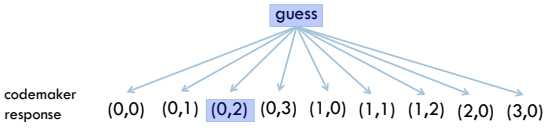
Playing

We selection one (the one with the smallest max)

What happens next?

Playing

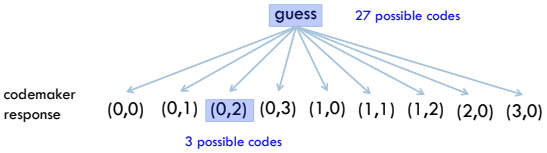
We make a guess and get a response



Now what?

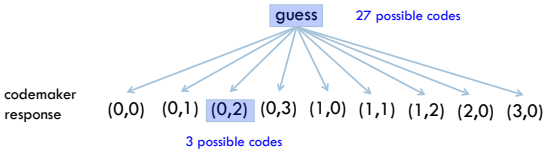
Playing

Just like before, we filter the list of possible candidates



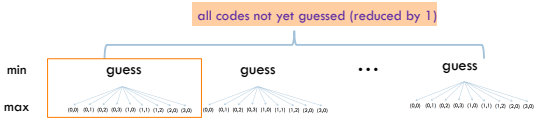
Playing

Just like before, we filter the list of possible candidates

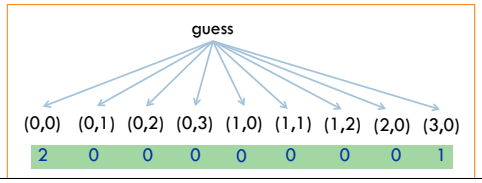


Now what?

Playing



We selection one (the one with the smallest max)



A more efficient solution

The one that minimizes the maximum remaining candidates

min [max [For all codes not yet guessed: Consider all possible responses: Calculate the size of the remaining candidates if we guessed that code and got that response] select response with largest remaining for that code] select code with smallest max

How expensive is this computation?

A more efficient solution

min [max [For all codes not yet guessed: Consider all possible responses: Calculate the size of the remaining candidates if we guessed that code and got that response] select response with largest remaining for that code] select code with smallest max

num_codes * num_responses * cost_to_calculate_remaining_size

= num_codes * num_responses * cost_to_filter_candidates

= num_codes * num_responses * current_remaining_candidates

How large is this at the top of the tree?

A more efficient solution

min [max [For all codes not yet guessed: Consider all possible responses: Calculate the size of the remaining candidates if we guessed that code and got that response] select response with largest remaining for that code] select code with smallest max

num_codes * num_responses * cost_to_calculate_remaining_size

= num_codes * num_responses * cost_to_filter_candidates

= num_codes * num_responses * current_remaining_candidates

1296 13 1296

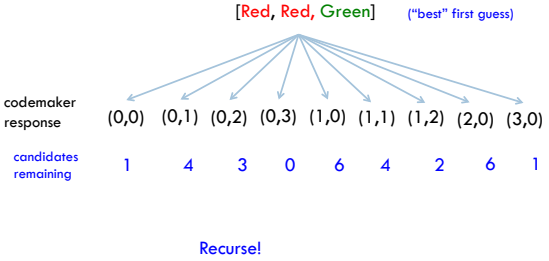
Game tree

We can precompute the entire tree of possibilities

Expensive upfront to compute

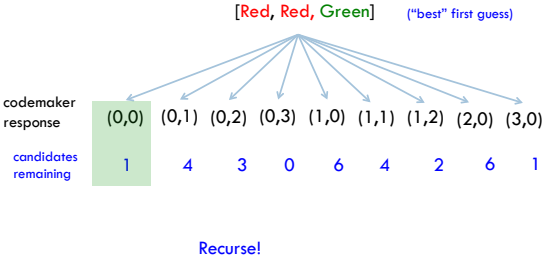
Playing, though, becomes fast

Game tree



Recurse!

Game tree



Recurse!

Game tree

Parent: [Red, Red, Green]

(0,0)

(26 guesses
1 candidate answer)

[Blue, Blue, Blue]

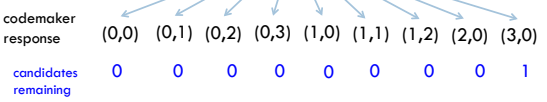
Game tree

Parent: [Red, Red, Green]

(0,0)

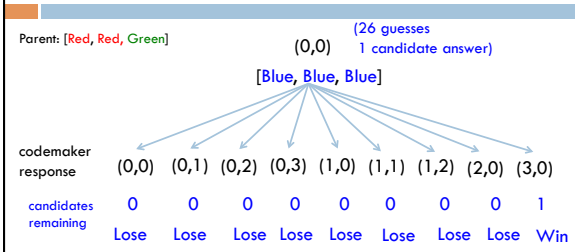
(26 guesses
1 candidate answer)

[Blue, Blue, Blue]



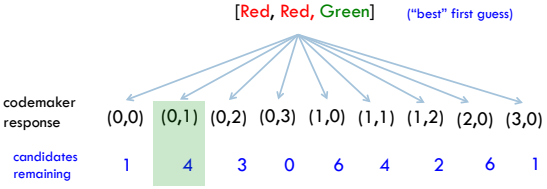
What now?

Game tree

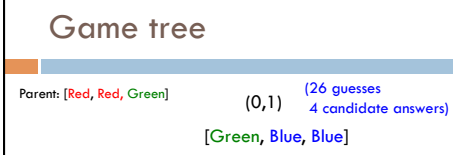


Use lose to indicate we don't have any options left (this shouldn't happen if we use a reasonable strategy)

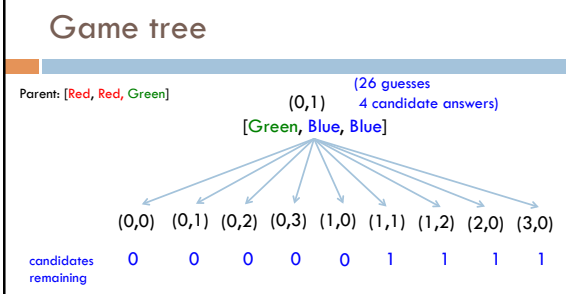
Game tree



Game tree

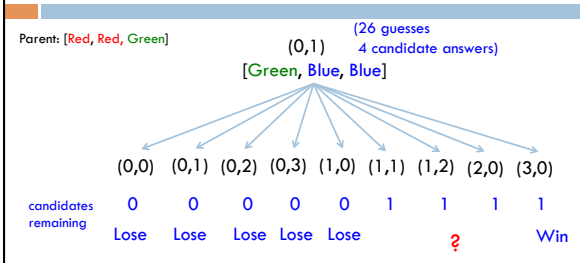


Game tree

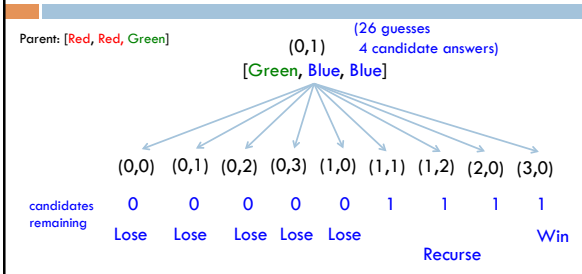


What now?

Game tree



Game tree



Building the game tree

If 0 options then Lose

If 1 option and the response ***was*** (num_pegs, 0) then Win

Otherwise, build another Tree:

- Guess = one that minimizes the maximum remaining candidates over all responses
  - Break ties by 1) those that are still valid codes and 2) found first in candidate (valid) list
- Recurse on responses