ADVERSARIAL SEARCH

David Kauchak
CS52 – Spring 2016

Admin

Assignment 6

Assignment 7

Game playing

Why study games?

In, why try and write computer programs that can play games?

Why study games

Clear success criteria

Good motivator to push research

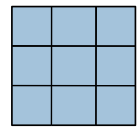
Important historically for AI

Fun ☺

Some real-world problems fit this model
- game theory (economics)
- multi-agent problems

Game playing results?

Tic Tac Toe



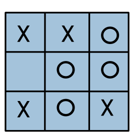
We want to write a program to play Tic Tac Toe.

How would you do it?

What types of decisions does it have to make?

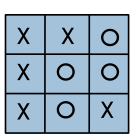
Tic Tac Toe

For any board configuration



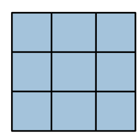
What move should 'X' make?

Needs to make a move (ideally as good as possible)



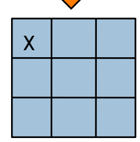
Tic Tac Toe

For any board configuration



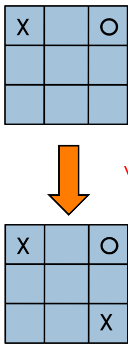
What move should 'X' make?

Needs to make a move (ideally as good as possible)



Tic Tac Toe

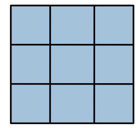
For any board configuration



What move should 'X' make?

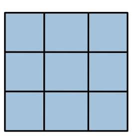
Needs to make a move (ideally as good as possible)

Tic Tac Toe



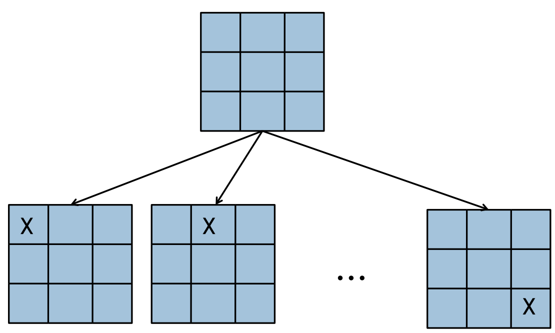
How can we "learn" this (or figure it out)?

Tic Tac Toe as search



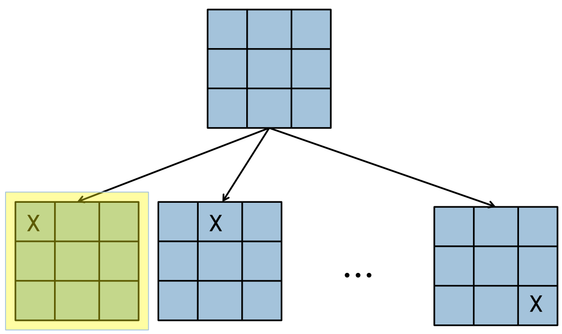
Key idea: search all possible moves/configurations

Tic Tac Toe as search



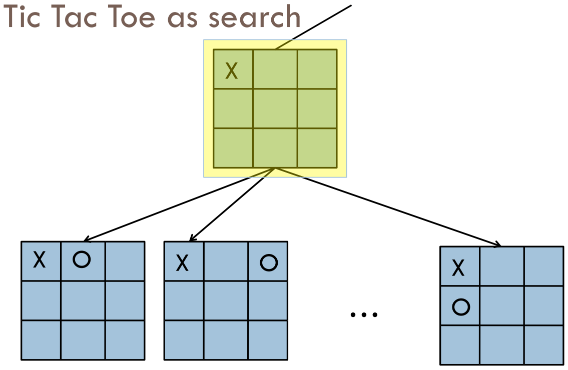
Explore all possible first moves by 'X'

Tic Tac Toe as search



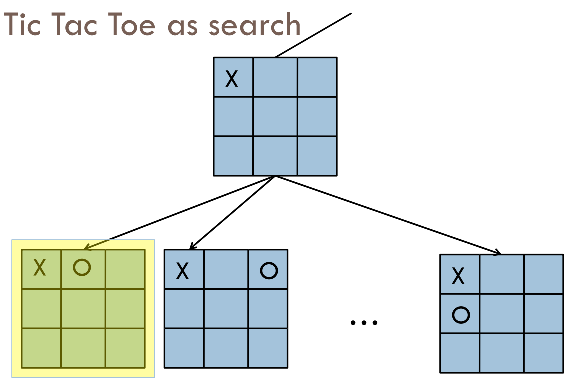
Then, explore all possible next moves by ‘O’

Tic Tac Toe as search



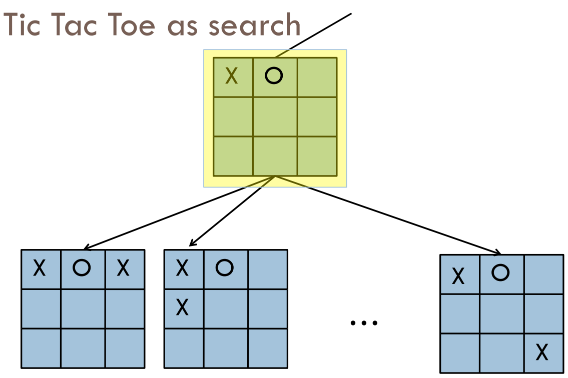
Explore all possible next moves by ‘O’

Tic Tac Toe as search



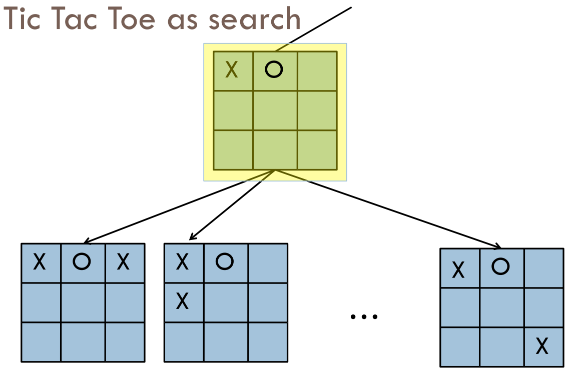
Then, explore all possible next moves by ‘X’

Tic Tac Toe as search



Explore all possible next moves by ‘X’

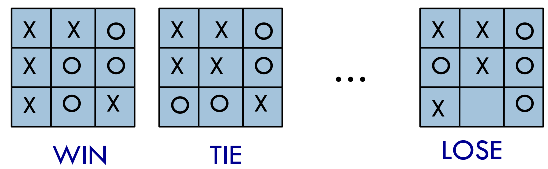
Tic Tac Toe as search



When do we stop?

Tic Tac Toe as search

Eventually, we'll get to an game end



How does this help us?

Try and make moves that move us towards a win, i.e. where there are leaves with a WIN.

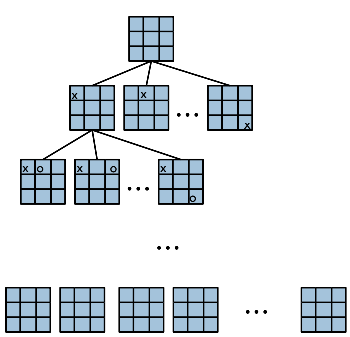
Tic Tac Toe

X's turn

O's turn

X's turn

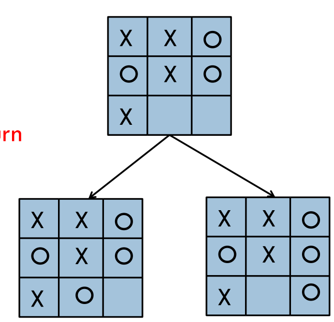
…



Problem: we don't know what 'O' will do

I'm X, what will 'O' do?

O's turn

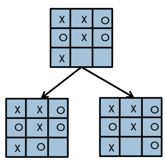


Minimizing risk

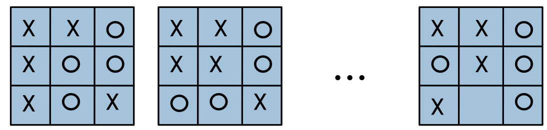
The program doesn't know what move O (the opponent) will make

It can *assume*, though, that it will try and make the **best move possible**

Even if O actually makes a different move, we're no worse off. Why?



Defining a scoring function



WIN +1

TIE 0

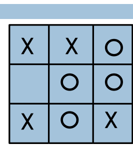
LOSE -1

Idea:

- define a function that gives us a "score" for how good each board state is for us
- higher scores mean better for us

Defining a scoring function

Our (X) turn

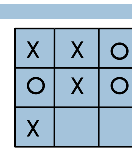


What should be the score of this board state?

+1: we can get to a win

Defining a scoring function

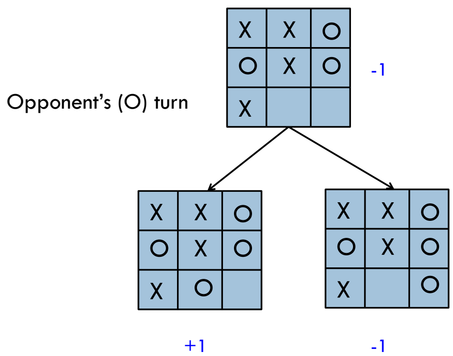
Opponent's (O) turn



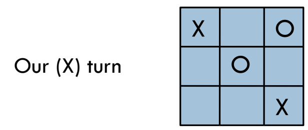
What should be the score of this board state?

-1: opponent can get to a win (we lose)

Defining a scoring function

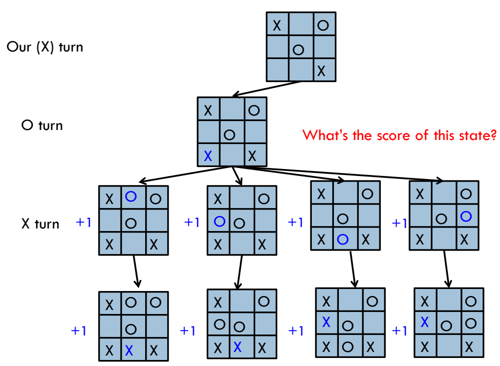


Defining a scoring function

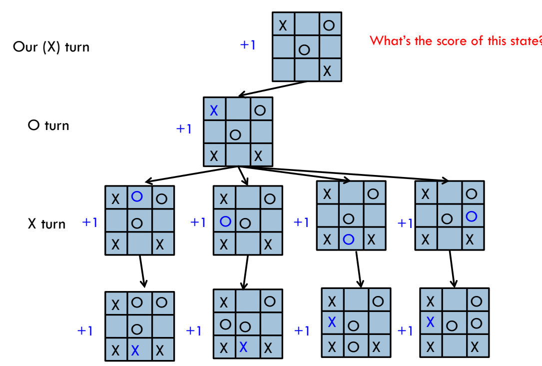


What should be the score of this board state?

Defining a scoring function

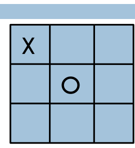


Defining a scoring function



Defining a scoring function

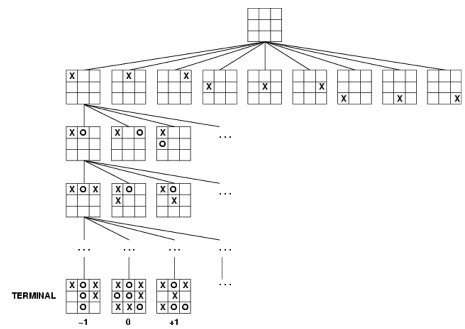
Our (X) turn



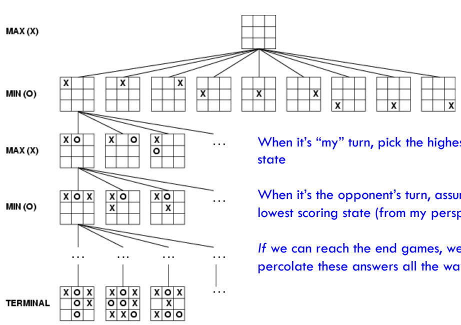
What should be the score of this state?

0: If we play perfectly and so does O, the best we can do is a tie (could do better if O makes a mistake)

How should X play?



How should X play?



When it's "my" turn, pick the highest scoring state

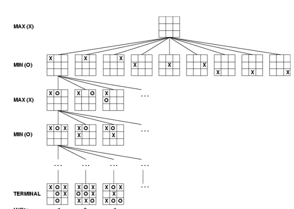
When it's the opponent's turn, assume the lowest scoring state (from my perspective)

If we can reach the end games, we can percolate these answers all the way back up

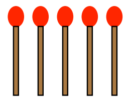
How to calculate this tree

Start from the leaves and propagate the score up:
- if X's turn, pick the move that maximizes the utility
- if O's turn, pick the move that minimizes the utility

This results in an optimal strategy!



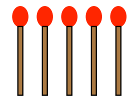
Another example: Baby Nim



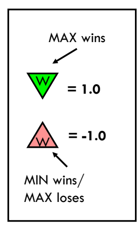
Take 1 or 2 sticks at each turn
Goal: take the last stick

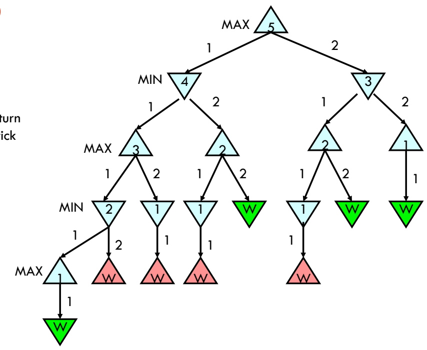
What move should I take?

Baby Nim

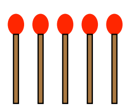


Take 1 or 2 at each turn
Goal: take the last stick

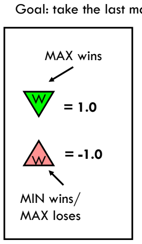


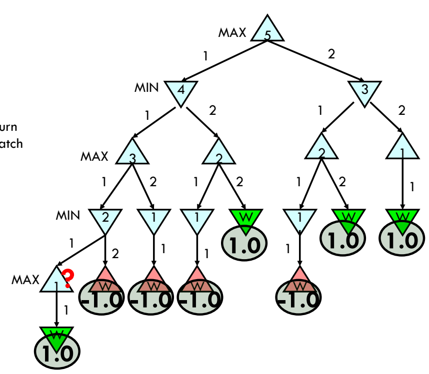


Baby Nim

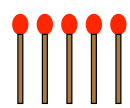


Take 1 or 2 at each turn
Goal: take the last match

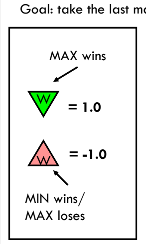


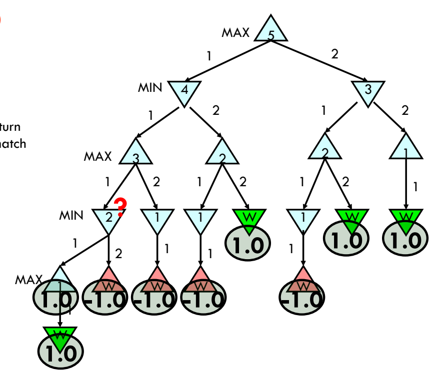


Baby Nim

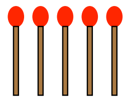


Take 1 or 2 at each turn
Goal: take the last match

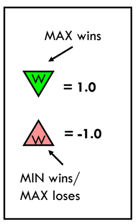


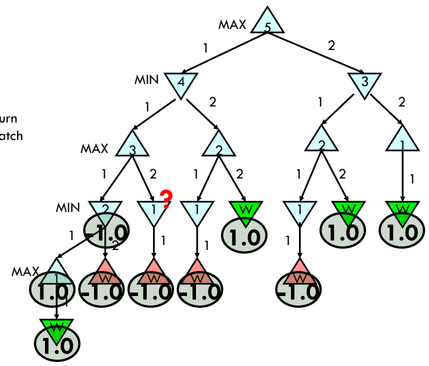


Baby Nim

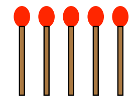


Take 1 or 2 at each turn
Goal: take the last match

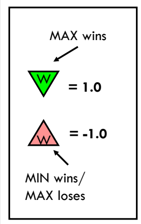


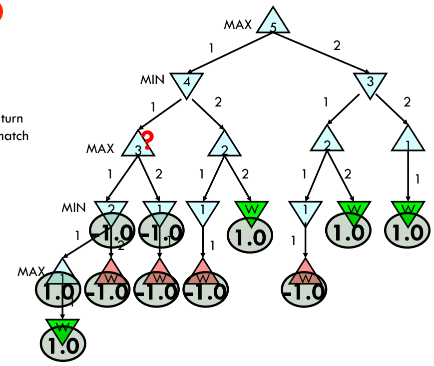


Baby Nim

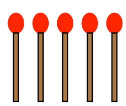


Take 1 or 2 at each turn
Goal: take the last match

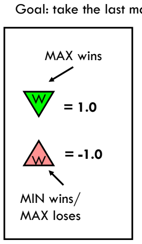


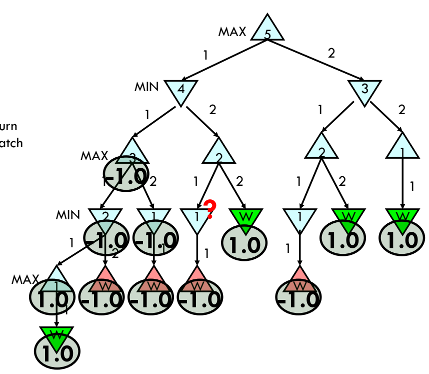


Baby Nim

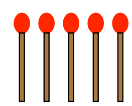


Take 1 or 2 at each turn
Goal: take the last match

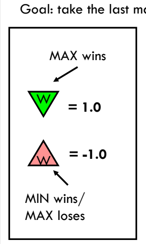


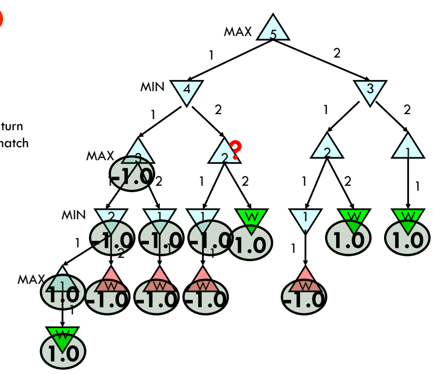


Baby Nim



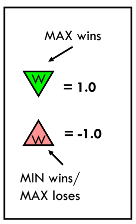
Take 1 or 2 at each turn
Goal: take the last match

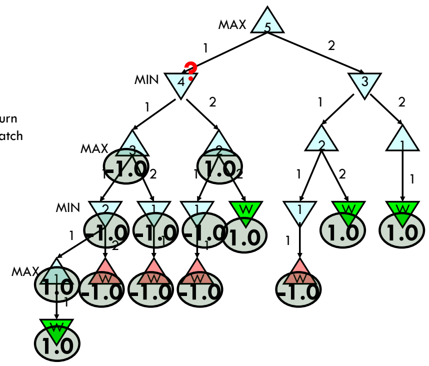




Baby Nim

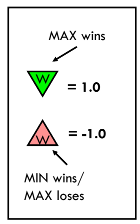
Take 1 or 2 at each turn
Goal: take the last match

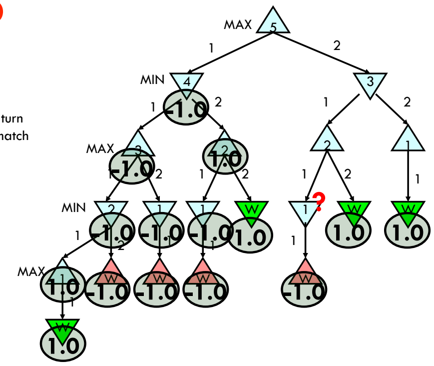




Baby Nim

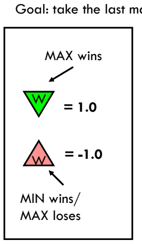
Take 1 or 2 at each turn
Goal: take the last match

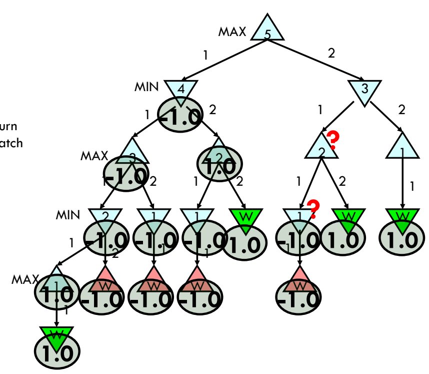




Baby Nim

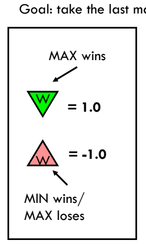
Take 1 or 2 at each turn
Goal: take the last match

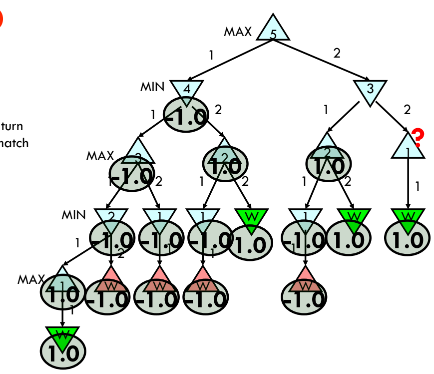




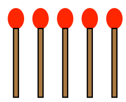
Baby Nim

Take 1 or 2 at each turn
Goal: take the last match

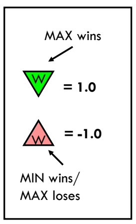


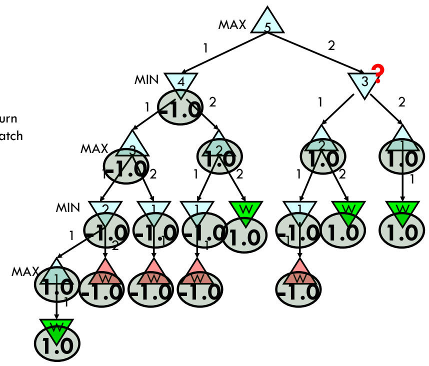


Baby Nim

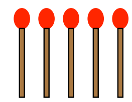


Take 1 or 2 at each turn
Goal: take the last match

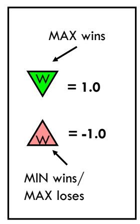


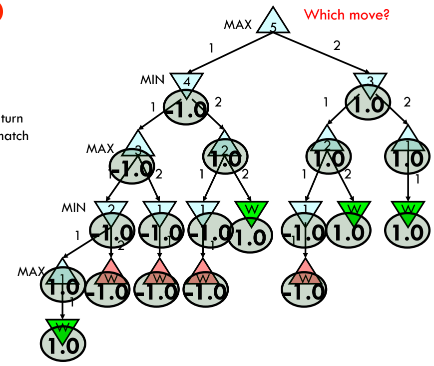


Baby Nim

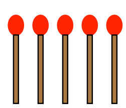


Take 1 or 2 at each turn
Goal: take the last match



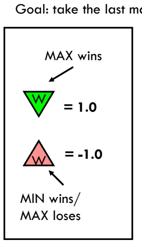


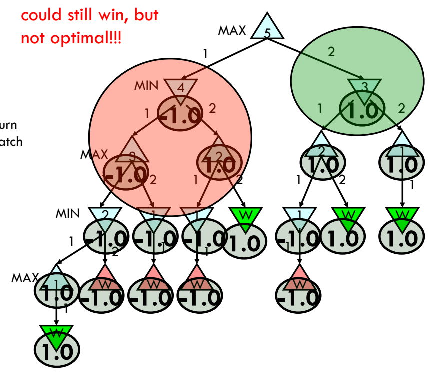
Baby Nim



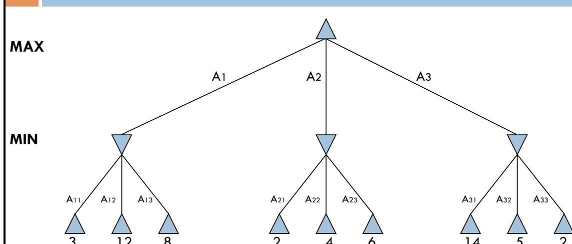
could still win, but
not optimal!!!

Take 1 or 2 at each turn
Goal: take the last match



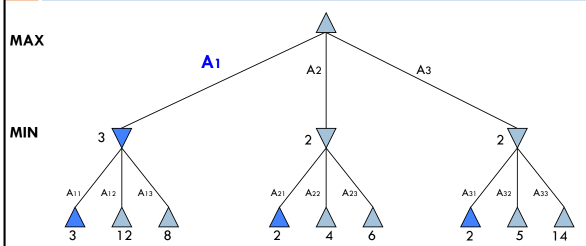


One more minimax example



Which move should be made: A_1, A_2 or A_3?

One more minimax example



Properties of minimax

Minimax is optimal!

What are the requirements to use minimax (i.e. what types of games can we solve)?

Properties of minimax

Minimax is optimal!

What are the requirements to use minimax (i.e. what types of games can we solve)?

- Must have full view of board state (e.g. stratego wouldn't work)
- No chance/probability
- Board state space can't be too big!

Game state size

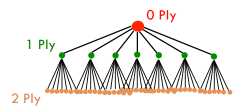
What impacts the size of the board state space?

Game state size

What impacts the size of the board state space?

- Number of possible moves from each board state
- Number of moves before the game finishes (depth of the tree)

Game state space size

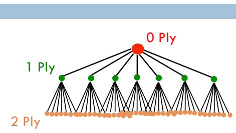


How many of this can we solve optimally?

Branching Factor Estimates
for different two-player games

Tic-tac-toe	4
Connect Four	7
Checkers	10
Othello	30
Chess	35
Go	300

Game state space size



Can search entire space

"solved" games

CHINOOK (2007)

Can't ☹

computer-dominated

Branching Factor Estimates
for different two-player games

Tic-tac-toe	4
Connect Four	7
Checkers	10
Othello	30
Chess	35
Go	300

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)

Exact Inexact

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

[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)

Exact Inexact

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

Exact Inexact

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

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

[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)

	Exact	Inexact
Guess 3: [Green, Blue, Blue] Response? (codemaker)	1	2

[Blue, Green, Blue]

[Blue, Blue, Green]

[Green, Blue, Blue]

Which ones can we eliminate?

Naïve approach (assignment 3)

	Exact	Inexact
Guess 3: [Green, Blue, Blue] Response? (codemaker)	1	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)

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

[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)?