Adversarial Search 2

CS51A
David Kauchak
Spring 2019

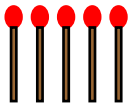
Some material borrowed from :
Sara Owsley Sood and others

Admin

Assignment 10

Pre-registration

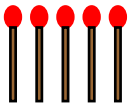
Baby Nim



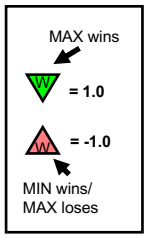
Take 1 or 2 at each turn
Goal: take the last match

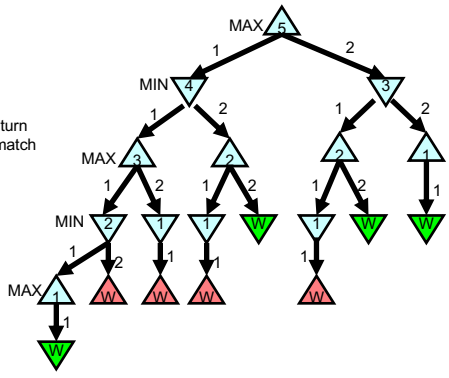
What move should I take?

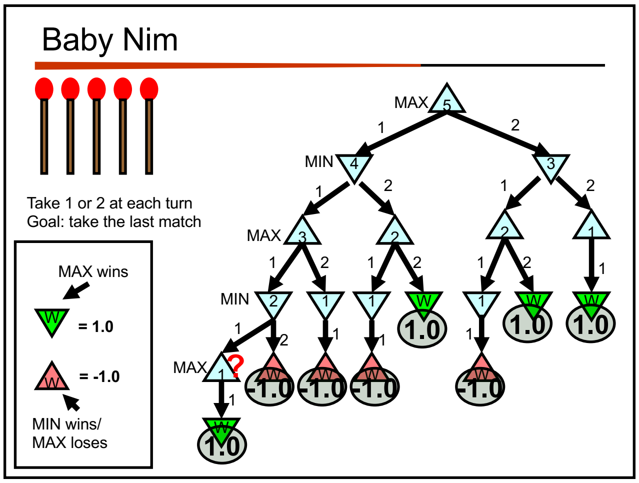
Baby Nim

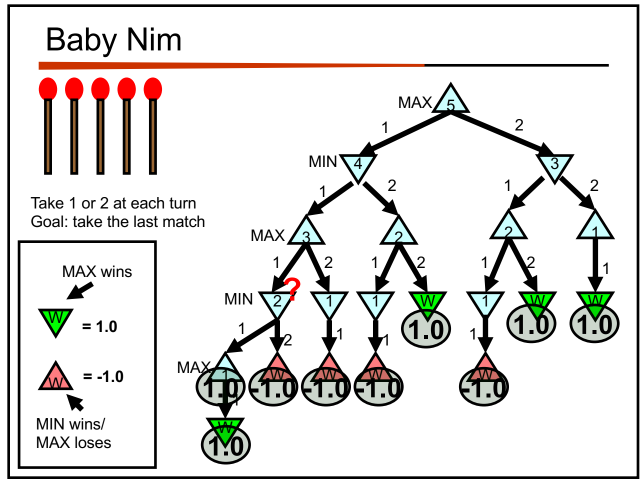


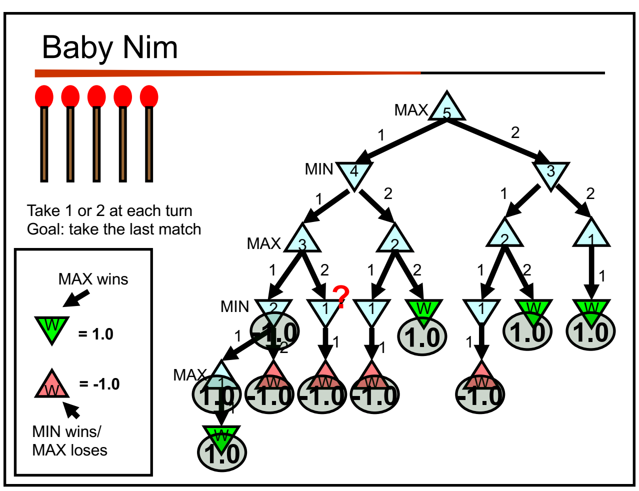
Take 1 or 2 at each turn
Goal: take the last match

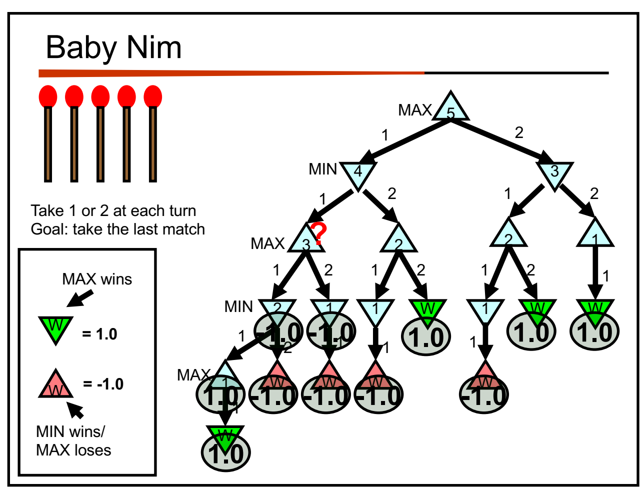




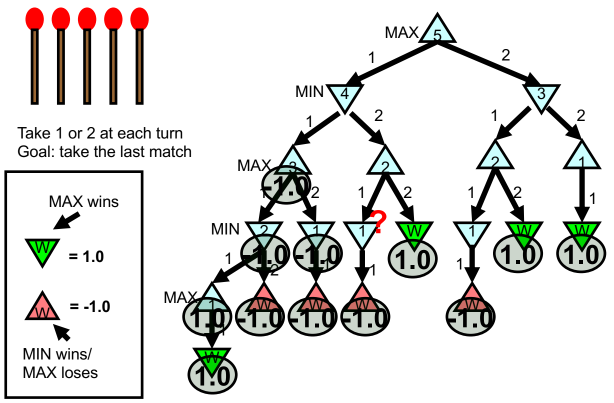

Baby Nim
Take 1 or 2 at each turn
Goal: take the last match
MAX wins
= 1.0
= -1.0
MIN wins/
MAX loses


Baby Nim
Take 1 or 2 at each turn
Goal: take the last match
MAX wins
= 1.0
= -1.0
MIN wins/
MAX loses

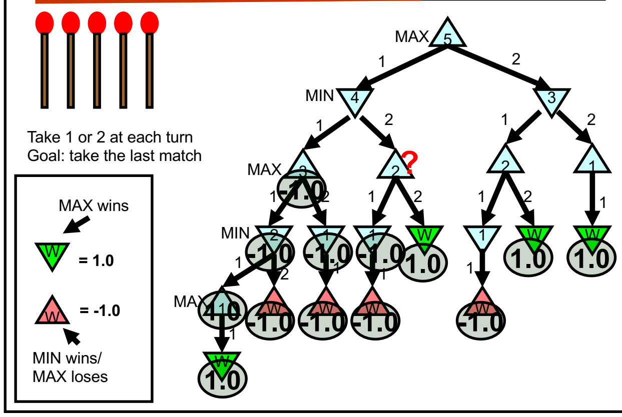

Baby Nim
Take 1 or 2 at each turn
Goal: take the last match
MAX wins
= 1.0
= -1.0
MIN wins/
MAX loses


Baby Nim
Take 1 or 2 at each turn
Goal: take the last match
MAX wins
= 1.0
= -1.0
MIN wins/
MAX loses

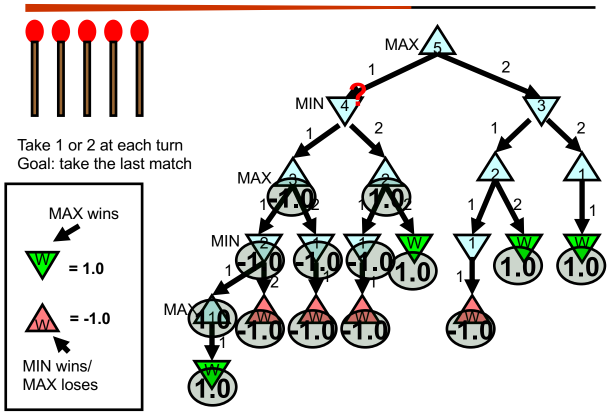
Baby Nim



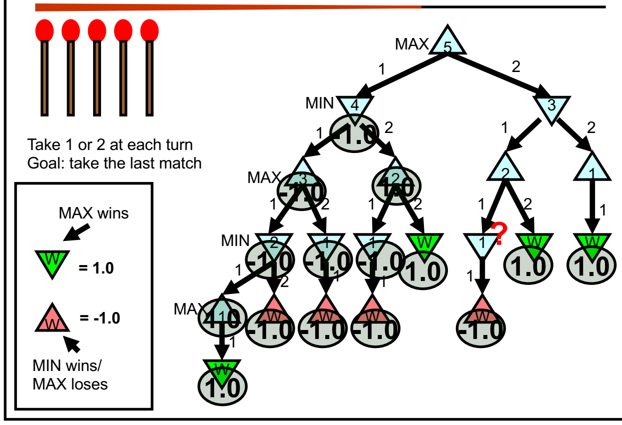
Baby Nim



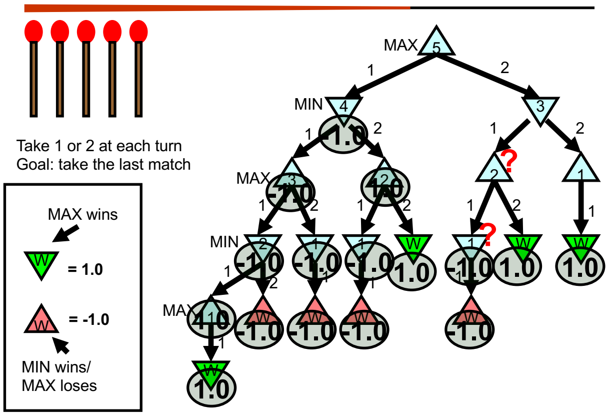
Baby Nim



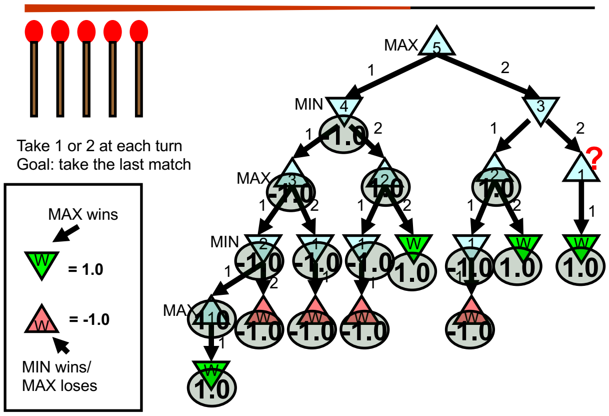
Baby Nim



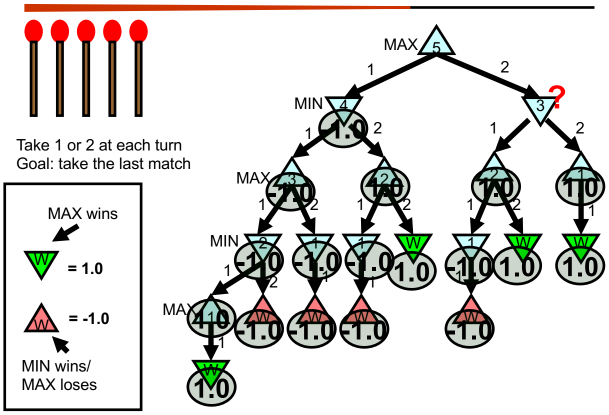
Baby Nim



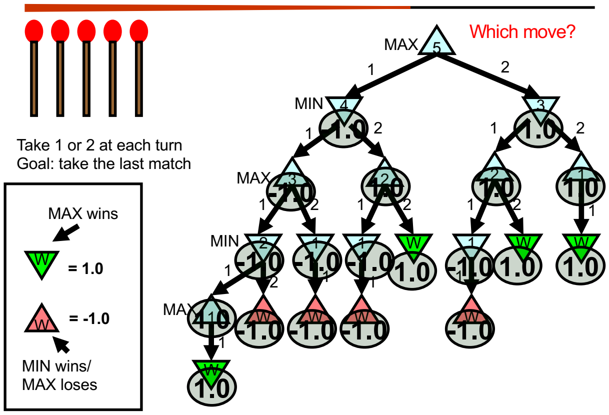
Baby Nim



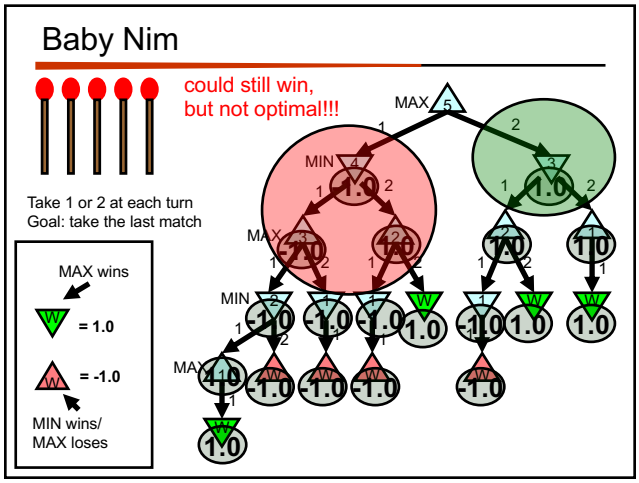
Baby Nim



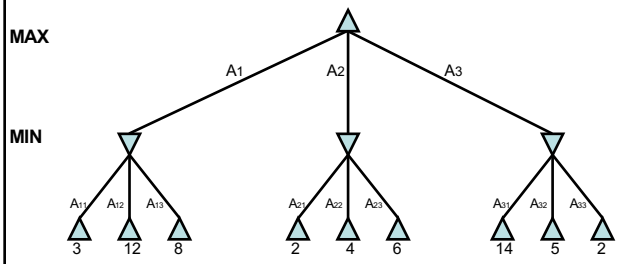
Baby Nim



Baby Nim

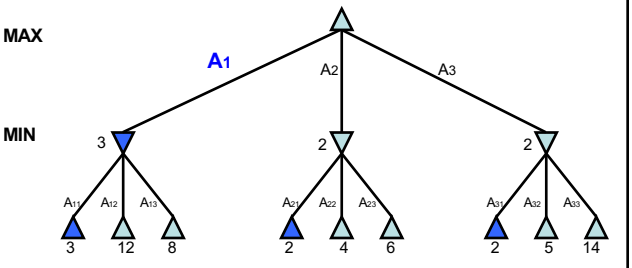


Minimax example 2



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

Minimax example 2



Properties of minimax

Minimax is optimal!

Are we done?

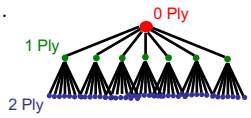
Games State Space Sizes

On average, there are ~35 possible moves that a chess player can make from any board configuration…

18 Ply!!

17005

Hydra at home in the United Arab Emirates…

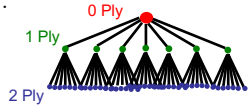


Branching Factor Estimates for different two-player games

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

Games State Space Sizes

On average, there are ~35 possible moves that a chess player can make from any board configuration…



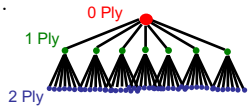
Boundaries for *qualitatively different games…*

Branching Factor Estimates for different two-player games

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

Games State Space Sizes

On average, there are ~35 possible moves that a chess player can make from any board configuration…



Can search entire space

"solved" games

CHINOOK (2007)

computer-dominated

Can't ⊗

Is this true? human-dominated

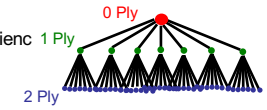
Branching Factor Estimates for different two-player games

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

Games State Space Sizes

AlphaGo (created by Google), in April 2016 beat one of the best Go players:

http://www.nytimes.com/2016/04/05/science/google-alphago-artificial-intelligence.html



Can search entire space

"solved" games

CHINOOK (2007)

computer-dominated

Can't ⊗

What do we do?

Branching Factor Estimates for different two-player games

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

Alpha-Beta pruning

An optimal pruning strategy

- only prunes paths that are suboptimal (i.e. wouldn't be chosen by an optimal playing player)
- returns the *same* result as minimax, but faster

Name the movie ☺

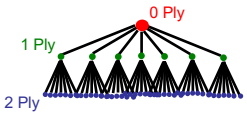
Alpha-Beta pruning

Games State Space Sizes

Pruning helps get a bit deeper

For many games, still can't search the entire tree

Now what?



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

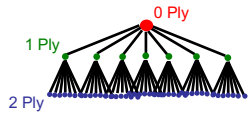
Games State Space Sizes

Pruning helps get a bit deeper

For many games, still can't search the entire tree

Go as deep as you can:

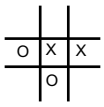
- *estimate* the score/quality of the state (called an evaluation function)
- use that instead of the real score



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

Tic Tac Toe evaluation functions



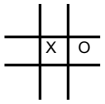
Ideas?

Example Tic Tac Toe EVAL

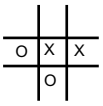
Tic Tac Toe
Assume MAX is using "X"

EVAL(*state*) =

if state is win for MAX:
+ ∞
if state is win for MIN:
- ∞
else:
(number of rows, columns and diagonals available to MAX) -
(number of rows, columns and diagonals available to MIN)

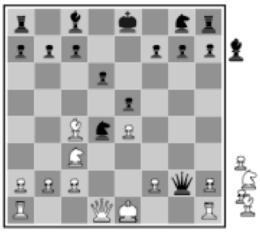


= 6 - 4 = 2



= 4 - 3 = 1

Chess evaluation functions

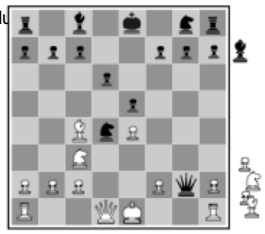


Ideas?

Chess EVAL

Assume each piece has the following valu
pawn = 1;
knight = 3;
bishop = 3;
rook = 5;
queen = 9;

EVAL(*state*) =
sum of the value of white pieces –
sum of the value of black pieces

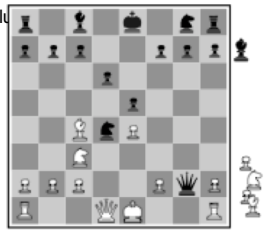


= 31 - 36 = -5

Chess EVAL

Assume each piece has the following valu
- pawn = 1;
- knight = 3;
- bishop = 3;
- rook = 5;
- queen = 9;

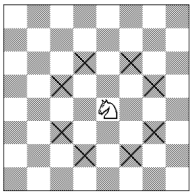
EVAL(*state*) =
sum of the value of white pieces –
sum of the value of black pieces



Any problems with this?

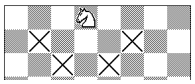
Chess EVAL

Ignores actual positions!



Actual heuristic functions are often a weighted combination of features

$$EVAL(s) = w_1 f_1(s) + w_2 f_2(s) + w_3 f_3(s) + \ldots$$



Chess EVAL

$$EVAL(s) = w_1 f_1(s) + w_2 f_2(s) + w_3 f_3(s) + \ldots$$

- f_1: number of pawns
- f_2: number of attacked knights
- f_3: 1 if king has knighted, 0 otherwise

A feature can be any numerical information about the board
- as general as the number of pawns
- to specific board configurations

Deep Blue: 8000 features!

history/end-game tables

History
- keep track of the quality of moves from previous games
- use these instead of search

end-game tables
- do a reverse search of certain game configurations, for example all board configurations with king, rook and king
- tells you what to do in ***any*** configuration meeting this criterion
- if you ever see one of these during search, you lookup exactly what to do

end-game tables

Devastatingly good

Allows much deeper branching
- for example, if the end-game table encodes a 20-move finish and we can search up to 14
- can search up to depth 34

Stiller (1996) explored all end-games with 5 pieces
- one case check-mate required 262 moves!

Knoval (2006) explored all end-games with 6 pieces
- one case check-mate required 517 moves!

Traditional rules of chess require a capture or pawn move within 50 or it's a stalemate

Opening moves

At the very beginning, we're the farthest possible from any goal state

People are good with opening moves

Tons of books, etc. on opening moves

Most chess programs use a database of opening moves rather than search

Nim

K piles of coins

On your turn you must take one or more coins from one pile

Player that takes the last coin wins

Example:
https://www.goobix.com/games/nim/