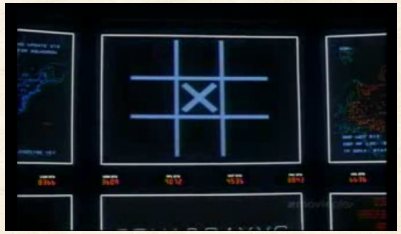
UNIVERSITY of WISCONSIN-MADISON
Computer Sciences Department

CS 202: Introduction to Computation

Professor Andrea Arpaci-Dusseau

How can computation… Win strategy games against you?



Intelligent Sheet of Paper

Today's Question

What type of algorithms are used to win strategy games?

Does computer need "insight" to win strategy games against humans?

Chess: Mechanical Turk

Automaton Chess Player

- Chess-playing machine 1770-1854
- Play strong game of chess against human opponent
- Arms move chess pieces
- Gears shown off inside

The Turk won most games

- Europe and the Americas
- Defeated many challengers (Napoleon Bonaparte and Benjamin Franklin)

The Turk: a mechanical illusion

- Human chess master hiding inside to operate the machine
- Revealed in 1820s

Chess: Deep Blue

Feb 1996: first machine to win chess game vs. reigning world champion
- Kasparov under regular time controls
- Deep Blues loses match

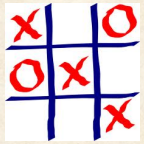
May 1997: Upgrade wins match
- Search 6-8 moves ahead (up to 20 moves)
- Kasparov said saw "deep intelligence and creativity" in machine's moves
  - Claimed person was directing Deep Blue
  - Change between games to fix weaknesses

What is a Strategy Game?

Requirements:
- No chance involved (no dice or card games)
- Both players have complete information
  - No hidden information (no Stratego or Magic)
- Two players alternate moves
  - No simultaneous moves. No races!
  - One player can pass…
- Identify ending condition as Win, Tie, or Lose:
  - Game ends in a pattern, capture, by the absence of moves

What is a Strategy Game?

Examples?
- Split (handgame), Tic-Tac-Toe, Connect 4, Othello, Checkers, Chess



Today's Exercise

Play strategy game to help you analyze your "strategy"

Want a strategy game you have no prior experience with… (not tic-tac-toe!)

Want you to blindly search for winning strategy
- Enumerate all possible moves
- Record whether each leads to win or loss

Exercise: Variation of Nim (Subtraction Game)

Rules:

- 2 players, 7 objects (in general, could be different numbers)
- Take turns removing 1, 2, or 3 objects
- Winner: Takes last object
- Strategy game: No chance, full info, take turns, identify winner

In order to record states:

- Fill in 7 slots (instead of remove)
- Use X (player 1)
- Use 0 (player 2)

Example: X, X000, X000XXX → X wins

Figure out: What is a winning strategy?

- Exhaustively enumerate possibilities until you find it…
- Hint: Player who goes first can always win

Exhaustively Analyze all Possibilities

7 Empty slots
Possibility 1: Initial move of 3 X's

Conclusion: X can always win if it places 3 (given 7 initially)
Leaving 4 squares is good…

Can first player win if they do something different?

Exhaustively Analyze all Possibilities

Possibility: Initial move of 2 X's

No matter what X does, O can win

X might not win if it takes 2 (given 7 initially)
Leaving 5 is bad…

How can we track all these options?

Use a "game tree"

- Very similar to decision tree
- Show choices that can be made by each player and the results

What should be at root of tree?

- Initial state
- Empty board…

Nim Game Trees

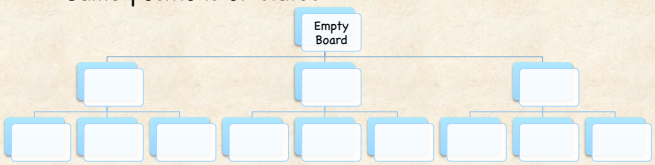
Root: Empty Boad

What do edges in game tree correspond to?

- Moves made by different players

What do nodes (boxes) of game tree show?

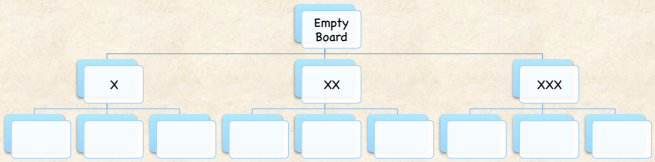
- Game positions or states



If X goes first, what are the next states?

Nim Game Trees

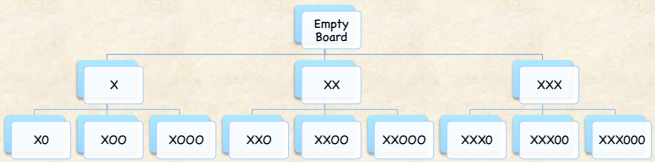
X has three possible moves: take 1, 2, or 3 objects



If 0 goes next, what are next nodes of tree?

Nim Game Trees

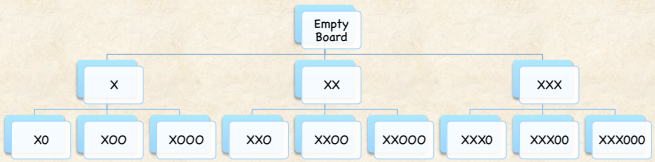
Enumerate all possible moves for 0 given past moves of X



Shows all possible boards after two moves

Nim Game Trees

Imagine extending Game Tree through all possible moves…

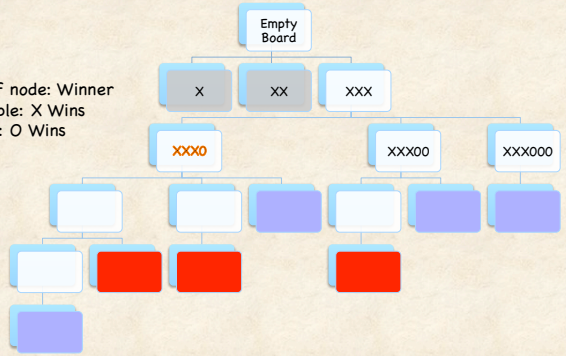


What will leaf nodes correspond to?

- Final game boards; winning or losing states

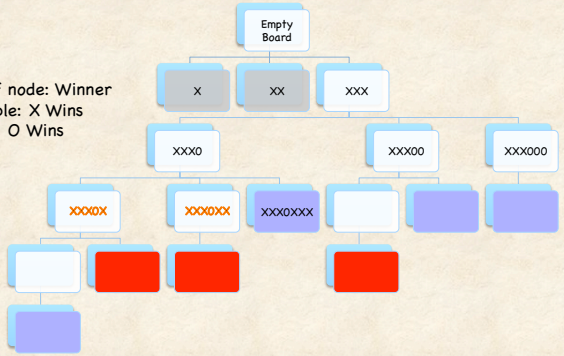
Nim Game Tree: Explore 3 X's for first move

Leaf node: Winner
Purple: X Wins
Red: O Wins



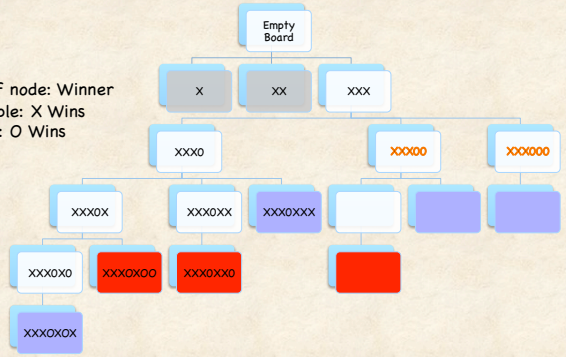
Nim Game Tree: 3 X's for first move

Leaf node: Winner
Purple: X Wins
Red: O Wins



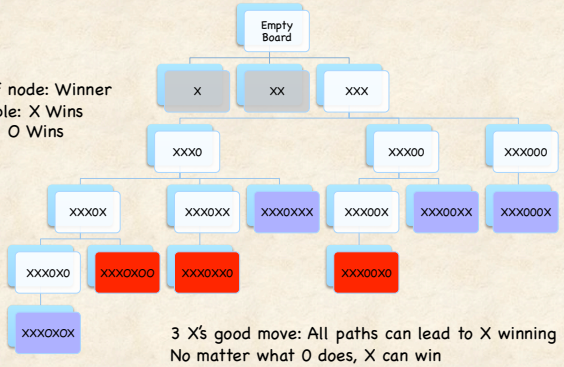
Nim Game Tree: 3 X's for first move

Leaf node: Winner
Purple: X Wins
Red: O Wins



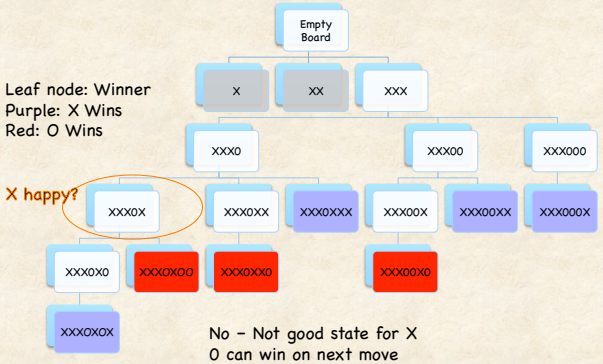
Nim Game Tree: 3 X's for first move

Leaf node: Winner
Purple: X Wins
Red: O Wins



3 X's good move: All paths can lead to X winning
No matter what O does, X can win
Always assume O is smart – will do best move for itself

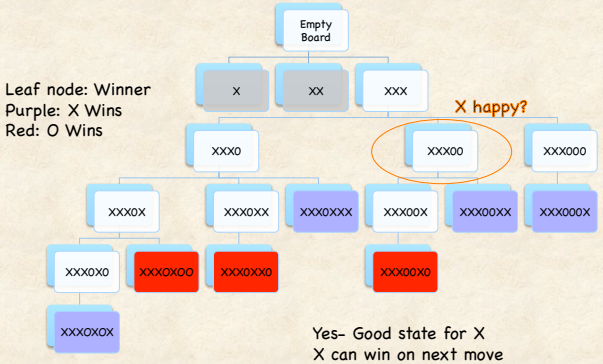
Nim Game Tree: 3 X's for first move



Leaf node: Winner
Purple: X Wins
Red: O Wins

No – Not good state for X
O can win on next move

Nim Game Tree: 3 X's for first move

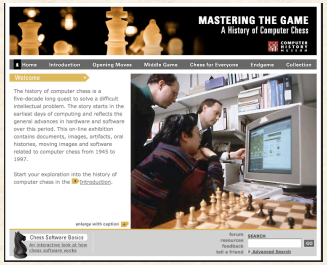


Leaf node: Winner
Purple: X Wins
Red: O Wins

Yes- Good state for X
X can win on next move

Need to develop more precise algorithm for traversing game trees

Computer Chess



Slideshow

http://spectrum.ieee.org/slideshow/computing/software/how-computer-chess-changed-programming

Great website

http://www.computerhistory.org/chess/

Interactive Demo:

http://www.computerhistory.org/chess/interact/index_content.html

Play the interactive demo…

Minimax Algorithm: Game Tree

Associate values:

- 0: tie
- 1: computer win
- -1: opponent win

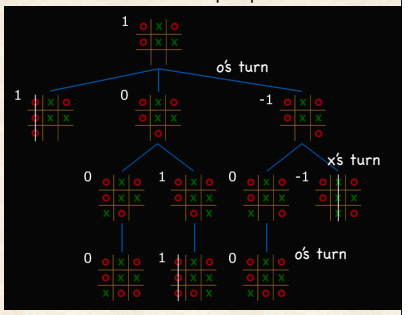
Strategy

- Assume each makes best move for itself
- Pick path to victory!

Algorithm

- Start at leaves
- Propagate max value before computer turn
- Propagate min value up before opponent turn
- Choose max path down

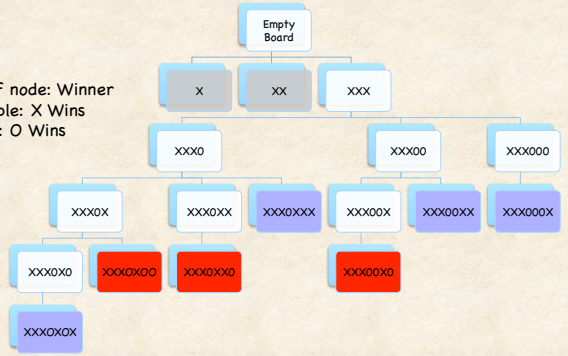
Calculate from o's perspective



O: Computer
X: Human
O's turn next…

Nim Game Tree: Minimax Algorithm

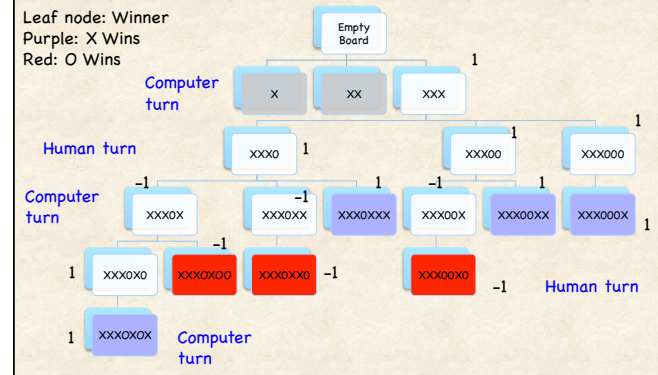
Leaf node: Winner
Purple: X Wins
Red: O Wins



3 X's good move: All paths can lead to X winning

Nim Game Tree: Minimax Algorithm

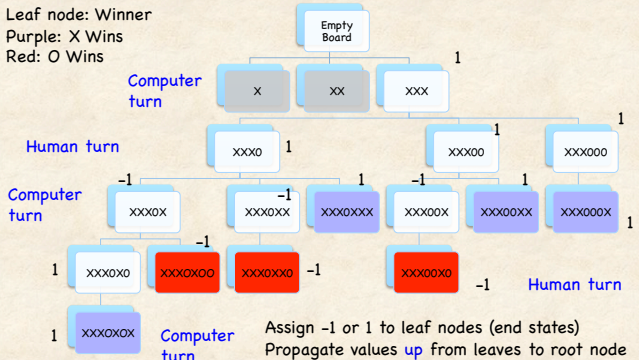
Leaf node: Winner
Purple: X Wins
Red: O Wins



Conclusion: Player 1 can always win with 7 objects by initially taking 3

Nim Game Tree: Minimax Algorithm

Leaf node: Winner
Purple: X Wins
Red: O Wins



Assign -1 or 1 to leaf nodes (end states)
Propagate values up from leaves to root node
- Use max when computer turn next (best move)
- Use min when human turn next (worst move)

Strongly Solve?

Nim and tic-tac-toe have relatively few board positions
- Can exhaustively search every possibility
- Can determine best strategy ahead-of-time and hard-code solution

Chess: too many board positions to exhaustively search
- Can only search few moves ahead and/or some possibilities at each move
- Determine strategy as play, based on observed positions

Check-Up

Which are true for the minimax algorithm?

- Uses psychology to guess what a human will do
- Uses probability to make the best decisions
- Used for games of chance
- Allows the computer to win every game
- Will find a winning move if it exists
- Assumes human opponent makes best possible moves
- Assigns "1" to end states where computer wins
- Propagates min value when it is human's turn next
- Can result in a huge number of states for complex games like chess
- Humans can use minimax algorithm

Announcements