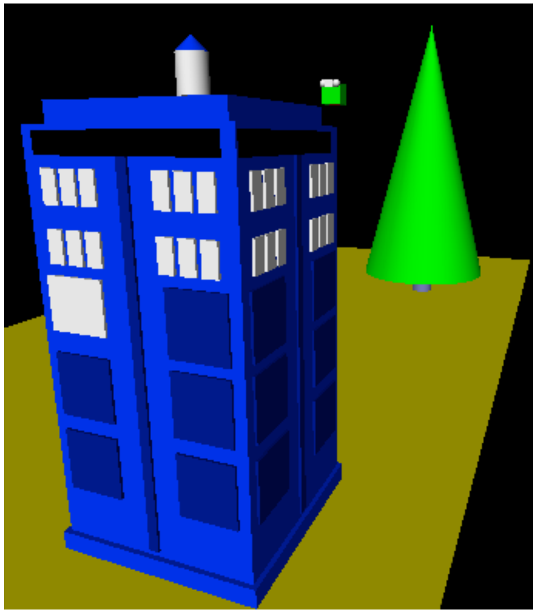
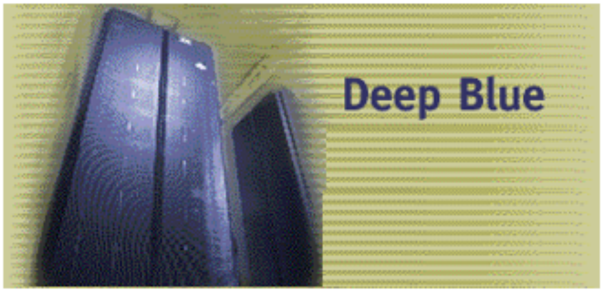
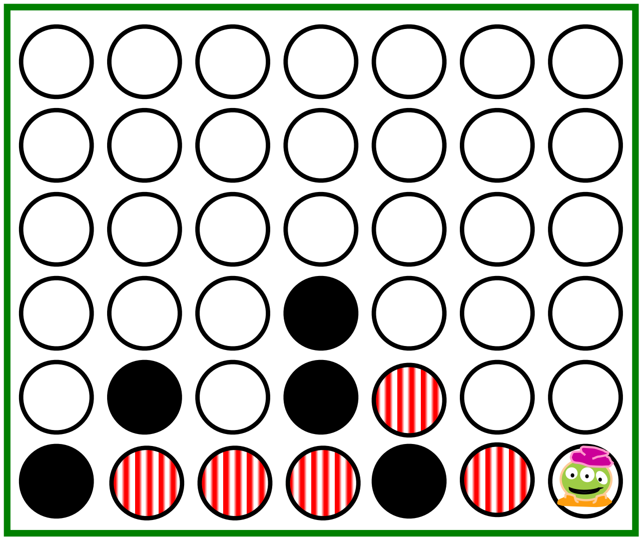
Intelligent CS 5 ?





Hw11 due **Monday** @ 11:59pm



b.playGame('human', o0rn), x3rn vs. 'human'

● X to move.

Is there a way to *ensure* a win?

If so, ***how far ahead***?

```
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | |X| | | |
| |X| |X|O| | |
|X|O|O|O|X|O| |
---------------
 0 1 2 3 4 5 6
```

Connect 4 AI ~ how *could* it work?

```
|X|X| | | | | |
|X|O| | |O| |X|
|O|O|X|O|X|O|O|
|X|X|X|O|O|X|X|
|O|O|O|X|O|O|X|
|O|X|X|X|O|X|O|
---------------
 0 1 2 3 4 5 6
```

Who won?!

It could just play randomly... *Let's try!*

Or, it could always play as far left as possible... *Let's try that, too!*

C4 AI ~ how *could* it work?

```
|X|X| | | | | |
|X|O| | |O| |X|
|O|O|X|O|X|O|O|
|X|X|X|O|O|X|X|
|O|O|O|X|O|O|X|
|O|X|X|X|O|X|O|
---------------
 0 1 2 3 4 5 6
```

```
while True:

    col = -1
    while b.allowsMove(col) == False:
        col = random.choice(range(7))

    b.addMove(ox, col)

    if ox == 'O':    ox = 'X'
    else:            ox = 'O'

    # check if game is over!
```

Who won?!

It could just play randomly... *Let's try!*

Or, it could always play as far left as possible... *Let's try that, too!*

C4 AI ~ how *could* it work?

```
|X|X| | | | | |
|X|O| | |O| |X|
|O|O|X|O|X|O|O|
|X|X|X|O|O|X|X|
|O|O|O|X|O|O|X|
|O|X|X|X|O|X|O|
---------------
 0 1 2 3 4 5 6
```

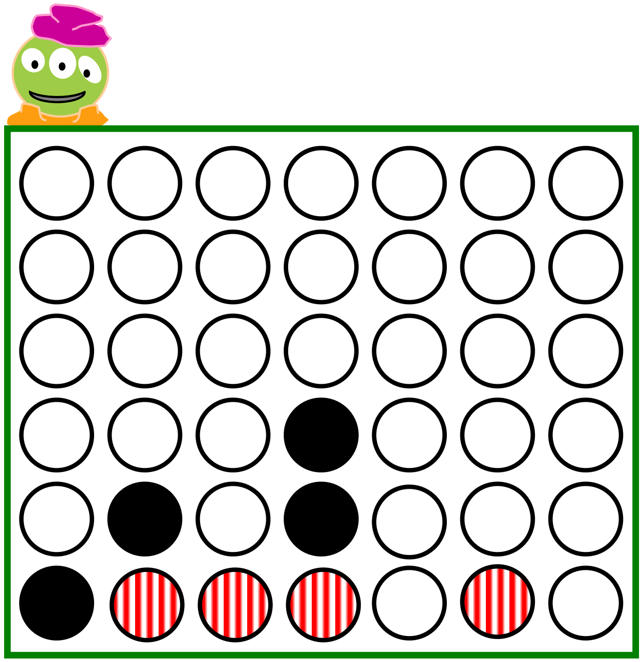
tiebreaking to the LEFT when possible...

```
|O|O|O| | | | |
|X|X|X| | | | |
|O|O|O| | | | |
|X|X|X| | | | |
|O|O|O| | | | |
|X|X|X|X| | | |
---------------
 0 1 2 3 4 5 6
```

It could just play randomly... *Let's try!*

Or, it could always play as far left as possible... *Let's try that, too!*

C4 AI ~ how ***should*** it work?



I feel ahead of the game here...

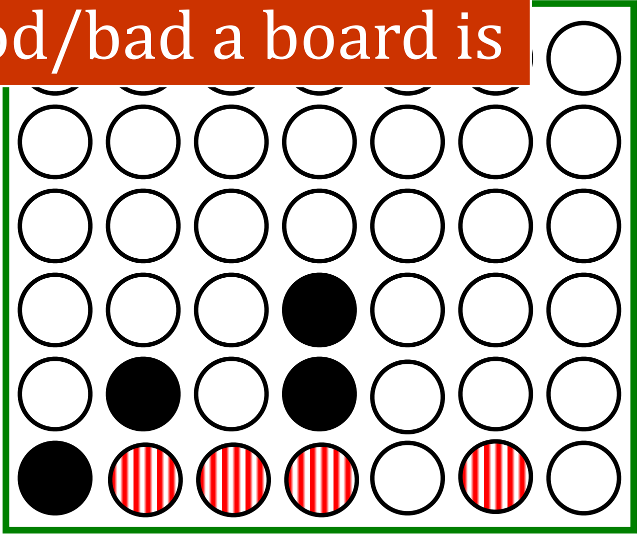
```
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | |X| | | |
| |X| |X| | | |
|X|O|O|O| |O| |
---------------
 0 1 2 3 4 5 6
```

It should **(1)** win and **(2)** *block* wins, if possible.

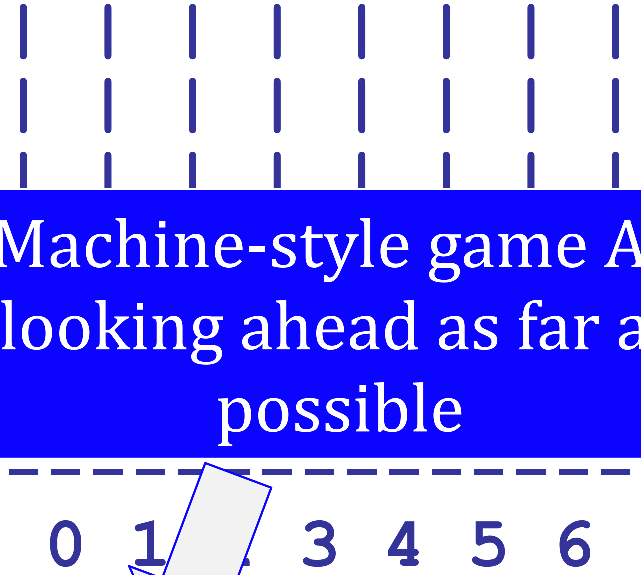
Otherwise it should just play as well as it can... ?!

C4 AI ~ how ***should*** it work?

Human-style game AI: "intuitive" evaluation of how good/bad a board is



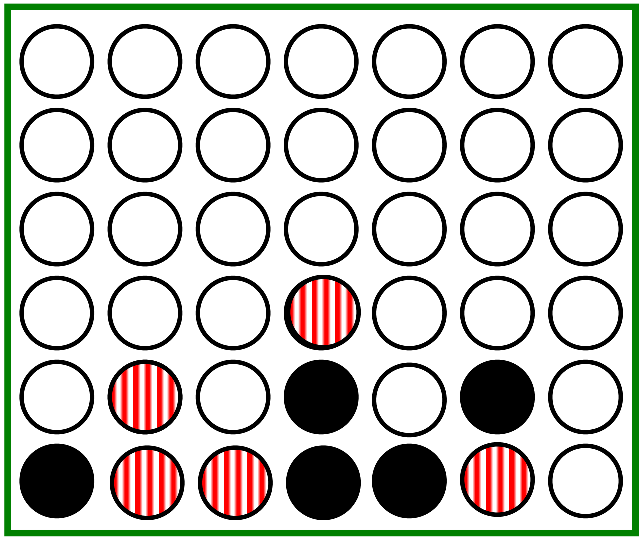
Machine-style game AI: looking ahead as far as possible



It should **(1)** win and **(2)** *block* wins, if possible.

Otherwise it should just play as well as it can... ?!

C4 AI ~ "intuitive" moves?

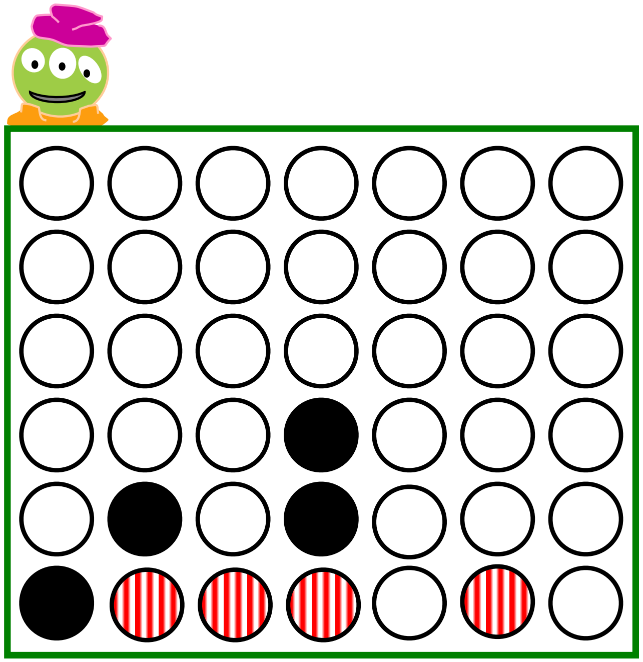


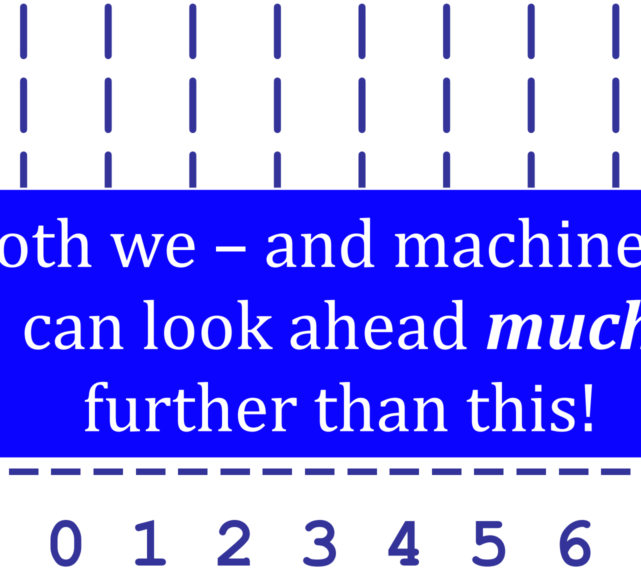
```
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | |O| | | |
| |O| |X| |X| |
|X|O|O|X|X|O| |
---------------
 0 1 2 3 4 5 6
```

If there isn't a win or loss... where should you go? Why?

C4 AI ~ *lookahead* moves...

I feel ahead of the game here...

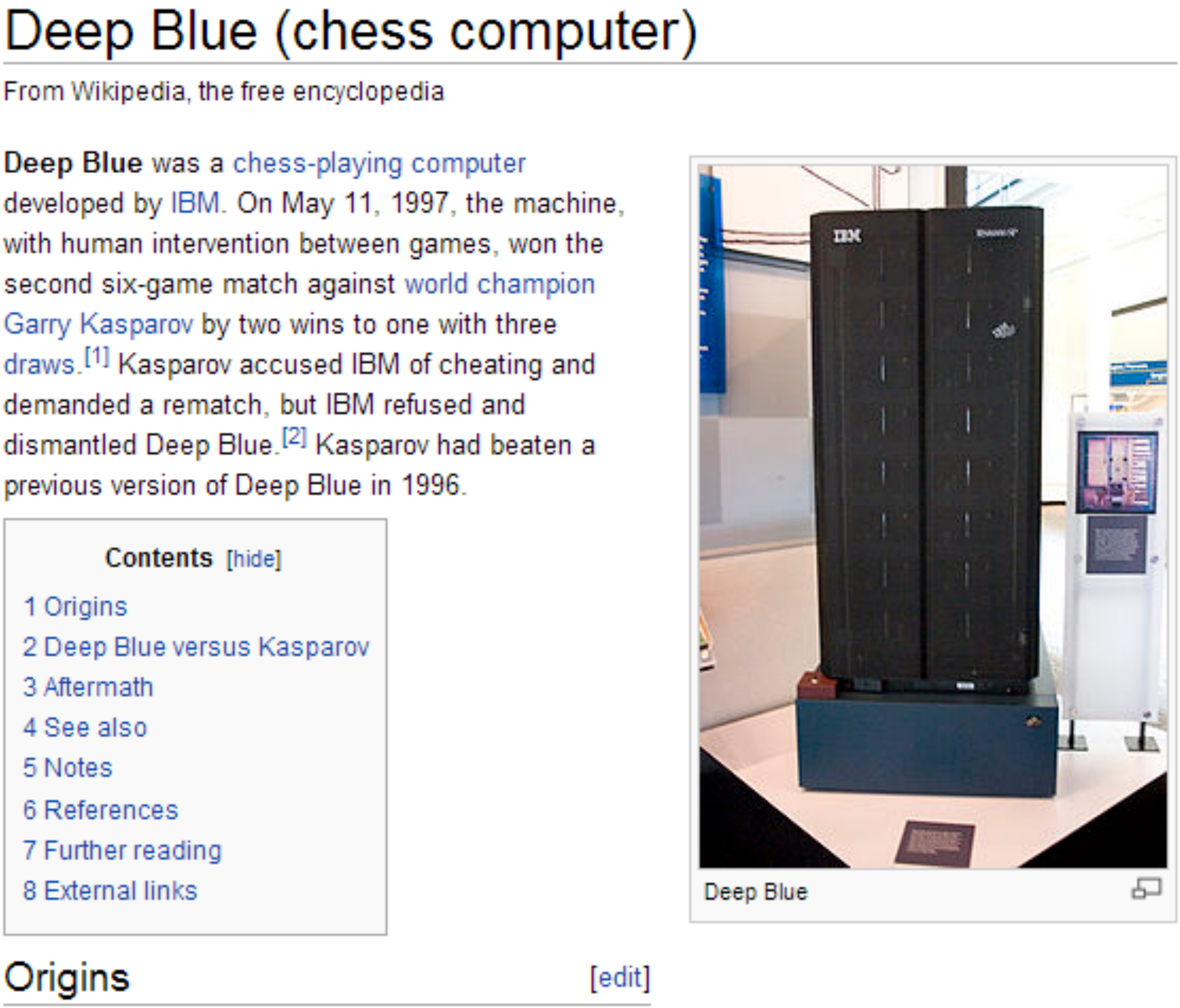




Both we – and machines – can look ahead ***much*** further than this!

It should **(1)** win and **(2)** *block* wins, when it can.

Otherwise it should just play as well as it can... ?!



Deep Blue (chess computer)

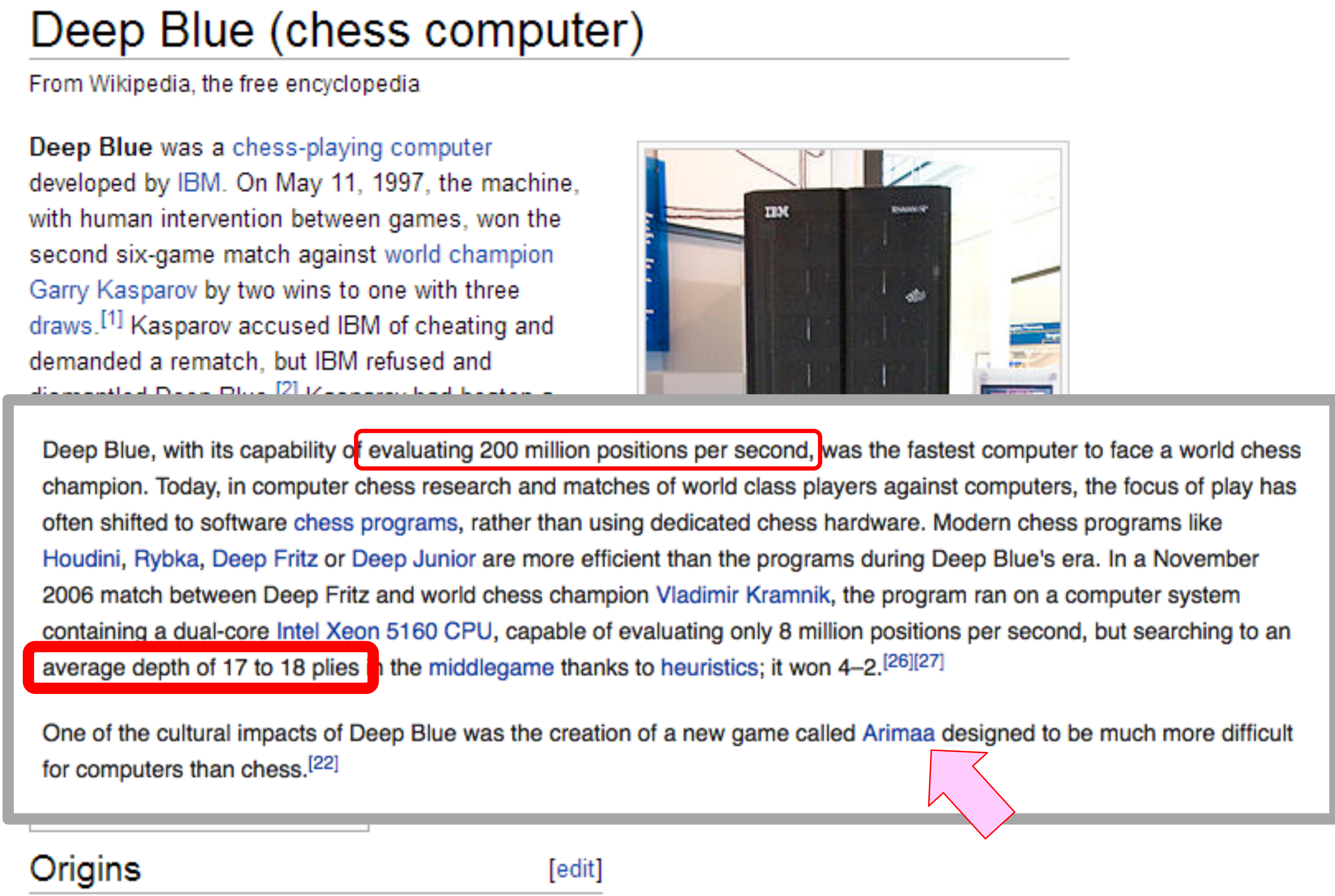
From Wikipedia, the free encyclopedia

Deep Blue was a chess-playing computer developed by IBM. On May 11, 1997, the machine, with human intervention between games, won the second six-game match against world champion Garry Kasparov by two wins to one with three draws.[1] Kasparov accused IBM of cheating and demanded a rematch, but IBM refused and dismantled Deep Blue.[2] Kasparov had beaten a previous version of Deep Blue in 1996.

Deep Blue

Contents [hide]

Origins [edit]



Deep Blue (chess computer)

From Wikipedia, the free encyclopedia

Deep Blue was a chess-playing computer developed by IBM. On May 11, 1997, the machine, with human intervention between games, won the second six-game match against world champion Garry Kasparov by two wins to one with three draws.[1] Kasparov accused IBM of cheating and demanded a rematch, but IBM refused and dismantled Deep Blue.[2] Kasparov had beaten a

Deep Blue, with its capability of evaluating 200 million positions per second, was the fastest computer to face a world chess champion. Today, in computer chess research and matches of world class players against computers, the focus of play has often shifted to software chess programs, rather than using dedicated chess hardware. Modern chess programs like Houdini, Rybka, Deep Fritz or Deep Junior are more efficient than the programs during Deep Blue's era. In a November 2006 match between Deep Fritz and world chess champion Vladimir Kramnik, the program ran on a computer system containing a dual-core Intel Xeon 5160 CPU, capable of evaluating only 8 million positions per second, but searching to an average depth of 17 to 18 plies in the middlegame thanks to heuristics; it won 4–2.[26][27]

One of the cultural impacts of Deep Blue was the creation of a new game called Arimaa designed to be much more difficult for computers than chess.[22]

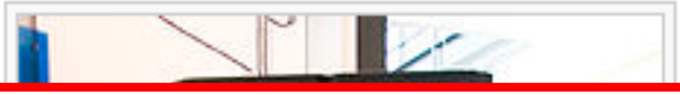
Origins

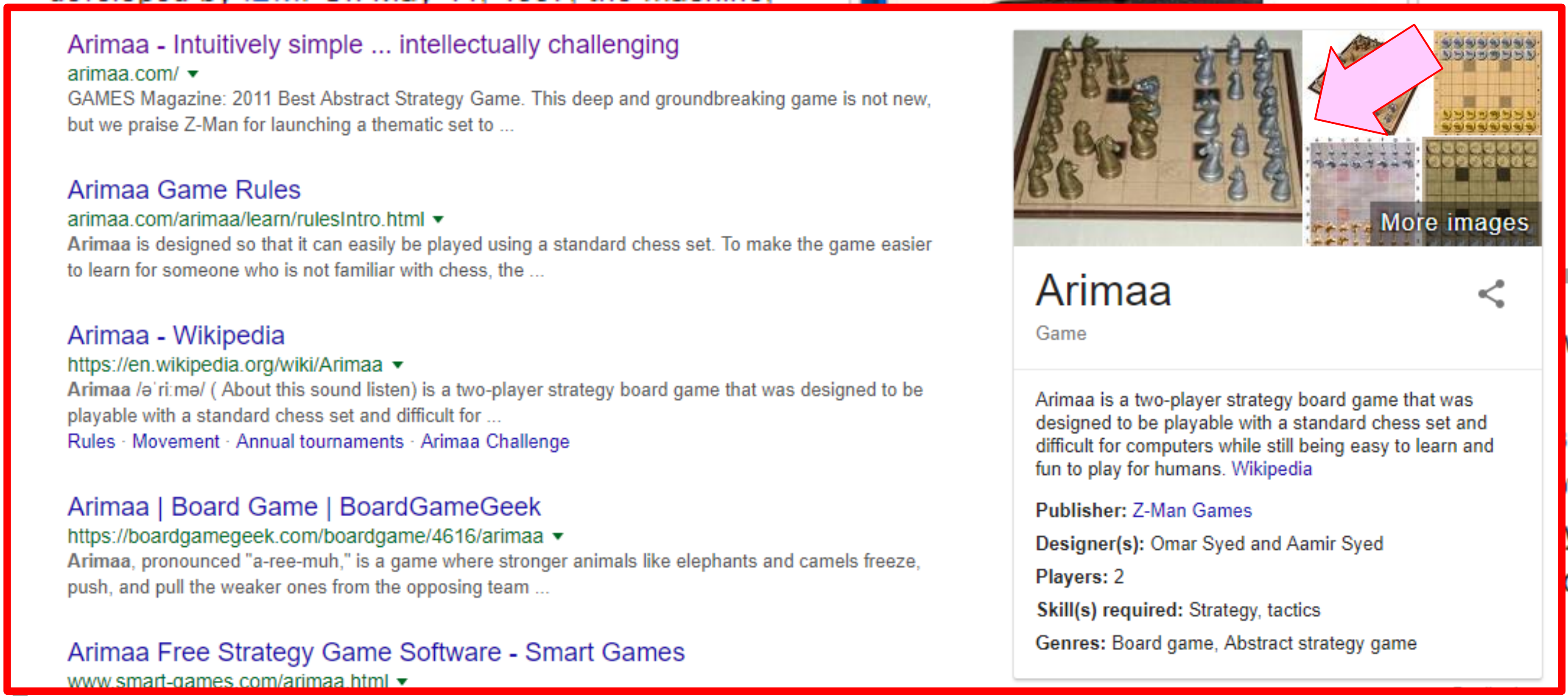
[edit]

Deep Blue (chess computer)

From Wikipedia, the free encyclopedia

Deep Blue was a chess-playing computer developed by IBM. On May 11, 1997, the machine,





Arimaa - Intuitively simple ... intellectually challenging
arimaa.com/ ▾
GAMES Magazine: 2011 Best Abstract Strategy Game. This deep and groundbreaking game is not new, but we praise Z-Man for launching a thematic set to ...

Arimaa Game Rules
arimaa.com/arimaa/learn/rulesIntro.html ▾
Arimaa is designed so that it can easily be played using a standard chess set. To make the game easier to learn for someone who is not familiar with chess, the ...

Arimaa - Wikipedia
https://en.wikipedia.org/wiki/Arimaa ▾
Arimaa /əˈriːmə/ (About this sound listen) is a two-player strategy board game that was designed to be playable with a standard chess set and difficult for ...
Rules · Movement · Annual tournaments · Arimaa Challenge

Arimaa | Board Game | BoardGameGeek
https://boardgamegeek.com/boardgame/4616/arimaa ▾
Arimaa, pronounced "a-ree-muh," is a game where stronger animals like elephants and camels freeze, push, and pull the weaker ones from the opposing team ...

Arimaa Free Strategy Game Software - Smart Games
www.smart-games.com/arimaa.html ▾

Arimaa

Game

Arimaa is a two-player strategy board game that was designed to be playable with a standard chess set and difficult for computers while still being easy to learn and fun to play for humans. Wikipedia

Publisher: Z-Man Games

Designer(s): Omar Syed and Aamir Syed

Players: 2

Skill(s) required: Strategy, tactics

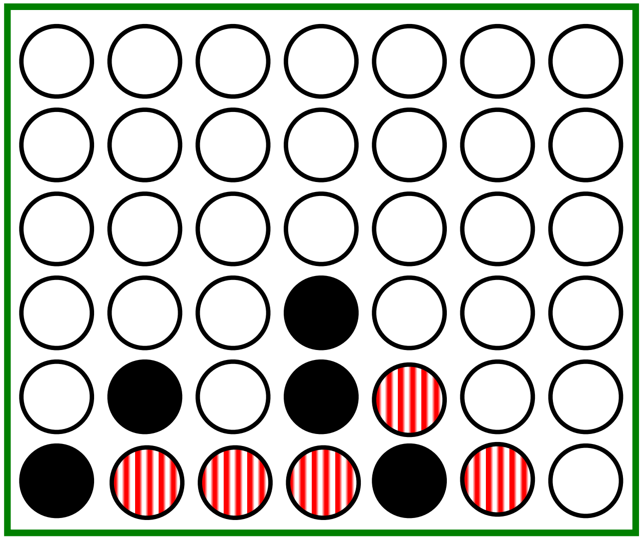
Genres: Board game, Abstract strategy game

world chess
of play has
like
vember
ystem
ching to an

One of the cultural impacts of Deep Blue was the creation of a new game called Arimaa designed to be much more difficult for computers than chess.[22]

Origins [edit]

Plies ~ *"turns to checkmate"* (for any game)



How many moves ahead might we have to look?

```
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | |X| | | |
| |X| |X|O| | |
|X|O|O|O|X|O| |
---------------
 0 1 2 3 4 5 6
```

```
b0

x5.scoresFor( b0 )
```

Plying our intuitions...

In all 4 of these boards, **X** will move to col 3, *even if both players tiebreak to the* **LEFT**

Find + circle *the reason* ***why*** *'X' moves to col. #3 for each...*

Name(s) ________________

bC

```
|X|O|O| | | | |
|X|X|X| | | | |
|O|X|O| | | | |
|O|X|X| | | | |
|O|O|O| | | | |
|X|X|O| | | | |
---------------
 0 1 2 3 4 5 6
```

ply == 0

bA

```
| | | | | | | |
| | | | | | | |
|O| | | | | | |
|X|O|X|O| | | |
|O|X|O|O| | | |
|X|X|O|X| | | |
---------------
 0 1 2 3 4 5 6
```

ply == 1

bD

```
| | | | | | | |
| | | | | | | |
|O|X| | | | | |
|X|X| | | | | |
|O|O|O| | | | |
|X|O|X|O| | | |
---------------
 0 1 2 3 4 5 6
```

ply == 2

bB

```
| | | | | | | |
| | | | | | | |
|X|O| | | | | |
|O|X| | | | | |
|O|O| | | | | |
|X|X| | | | | |
---------------
 0 1 2 3 4 5 6
```

ply == 3

```
|O|O|O| | | | |
|X|X|X| | | | |
|O|O|O| | | | |
|X|X|X| | | | |
|O|O|O| | | | |
|X|X|X|X| | | |
---------------
 0 1 2 3 4 5 6
```

Example

X: ply == 0,
O: ply == 0,
both: tbt == 'LEFT'

b.playGame(x0, o0)

Challenge: What will happen if you run **X at 1 ply** and **O at 1 ply**, each tiebreaking LEFT?

Challenge: What about **2-ply** each?

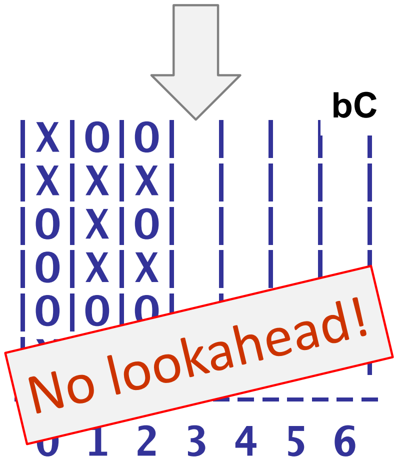
```
|O|O| | | | | |
|X|X| | | | | |
|O|O| | | | | |
|X|X| | | | | |
|O|O| | | | | |
|X|X| | | | | |
---------------
 0 1 2 3 4 5 6
```

Plying our intuitions...

In all 4 of these boards, **X** will move to col 3, *even if both players tiebreak to the* **LEFT**

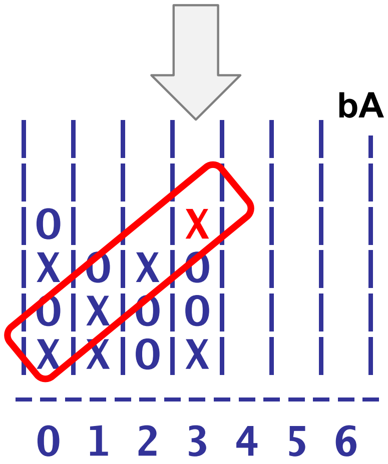
Find + circle *the reason* ***why*** *'X' moves to col. #3 for each...*

Try this on the back page first...



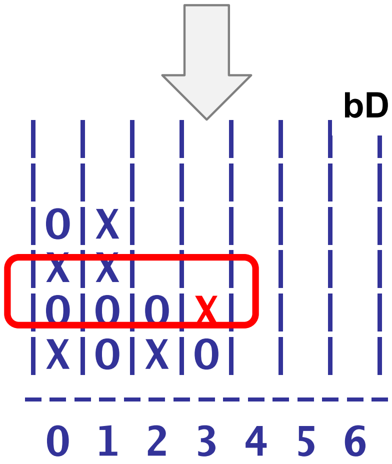
```
|X|O|O| | | | |
|X|X|X| | | | |
|O|X|O| | | | |
|O|X|X| | | | |
|O|O|O|
---------------
 0 1 2 3 4 5 6
```

ply == 0



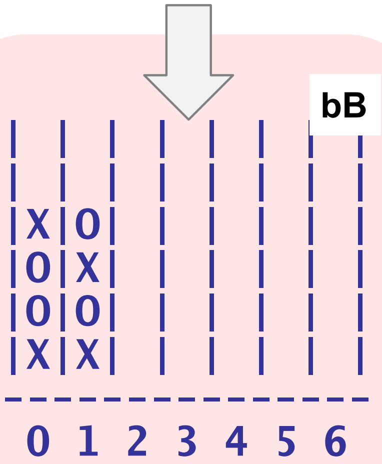
```
| | | | | | | |
| | | | | | | |
|O| | |X| | | |
|X|O|X|O| | | |
|O|X|O|O| | | |
|X|X|O|X| | | |
---------------
 0 1 2 3 4 5 6
```

ply == 1



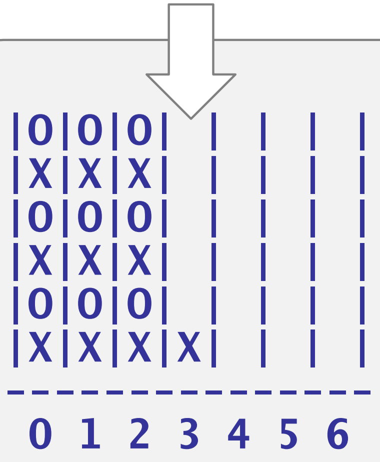
```
| | | | | | | |
| | | | | | | |
|O|X| | | | | |
|X|X| | | | | |
|O|O|O|X| | | |
|X|O|X|O| | | |
---------------
 0 1 2 3 4 5 6
```

ply == 2



```
| | | | | | | |
| | | | | | | |
|X|O| | | | | |
|O|X| | | | | |
|O|O| | | | | |
|X|X| | | | | |
---------------
 0 1 2 3 4 5 6
```

ply == 3



```
|O|O|O| | | | |
|X|X|X| | | | |
|O|O|O| | | | |
|X|X|X| | | | |
|O|O|O| | | | |
|X|X|X|X| | | |
---------------
 0 1 2 3 4 5 6
```

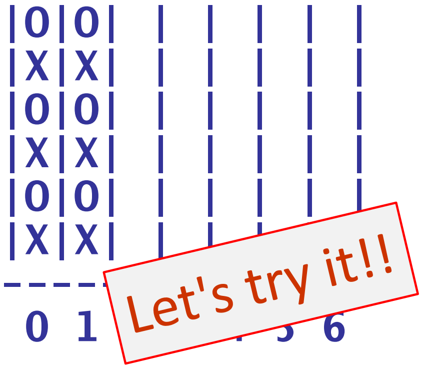
Example

X: ply == 0,
O: ply == 0,
both: tbt == 'LEFT'

b.playGame(x0, o0)

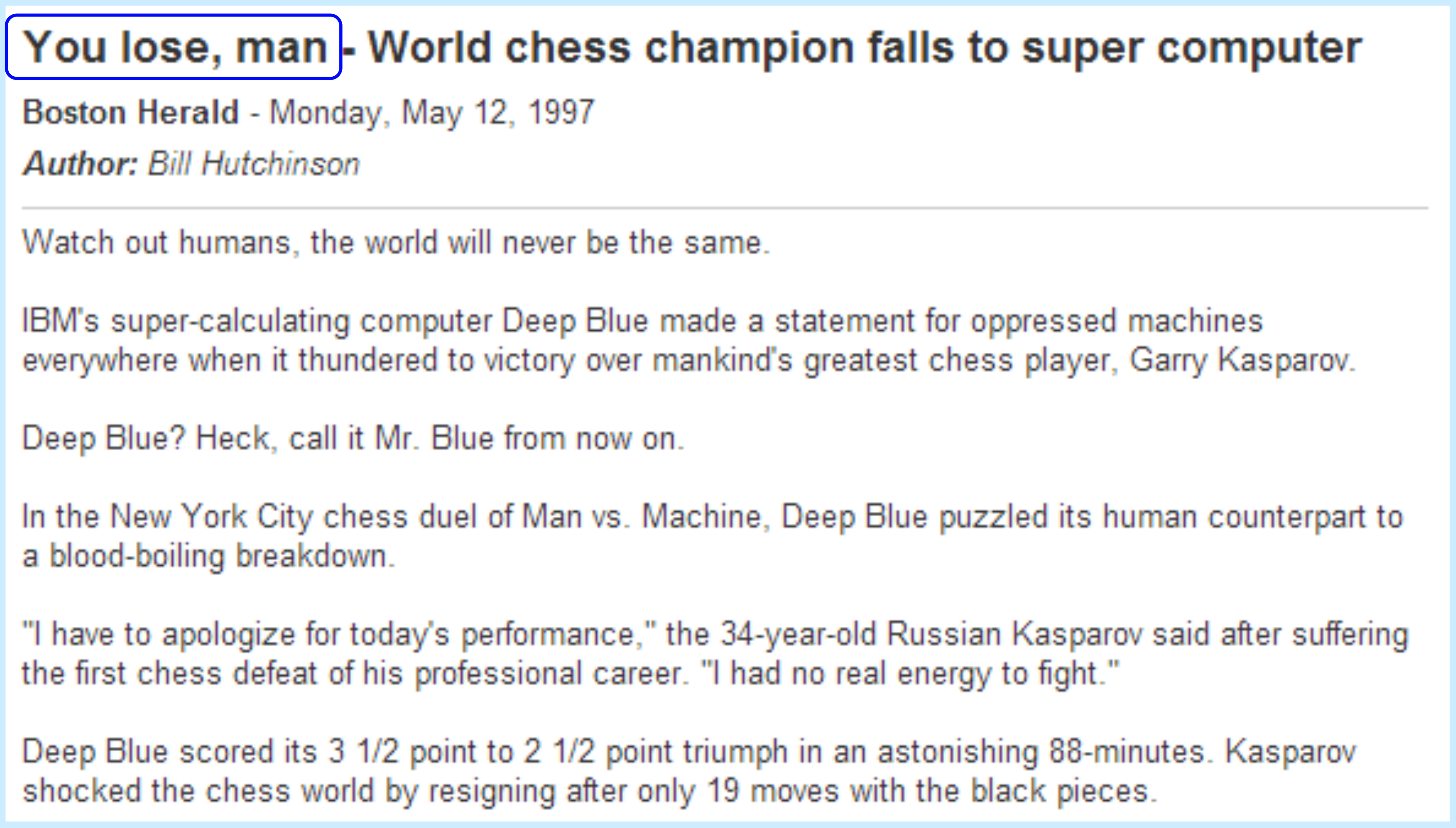
Challenge: What will happen if you run **X at 1 ply** and **O at 1 ply**, each tiebreaking LEFT?

Challenge: What about **2-ply** each?



```
|O|O| | | | | |
|X|X| | | | | |
|O|O| | | | | |
|X|X| | | | | |
|O|O| | | | | |
|X|X| | | | | |
---------------
 0 1 2 3 4 5 6
```

After Deep Blue…



You lose, man - World chess champion falls to super computer

Boston Herald - Monday, May 12, 1997

Author: *Bill Hutchinson*

Watch out humans, the world will never be the same.

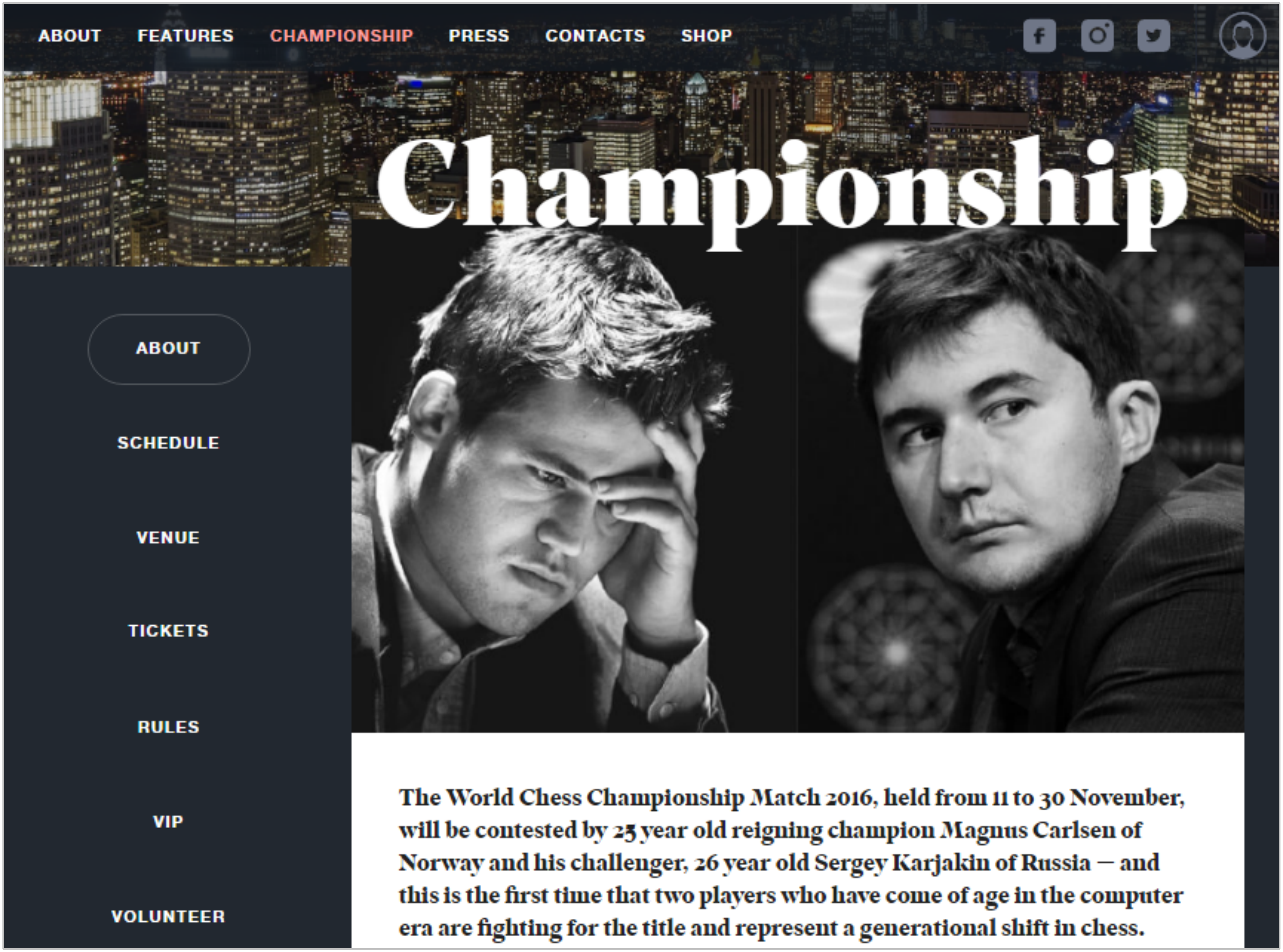
IBM's super-calculating computer Deep Blue made a statement for oppressed machines everywhere when it thundered to victory over mankind's greatest chess player, Garry Kasparov.

Deep Blue? Heck, call it Mr. Blue from now on.

In the New York City chess duel of Man vs. Machine, Deep Blue puzzled its human counterpart to a blood-boiling breakdown.

"I have to apologize for today's performance," the 34-year-old Russian Kasparov said after suffering the first chess defeat of his professional career. "I had no real energy to fight."

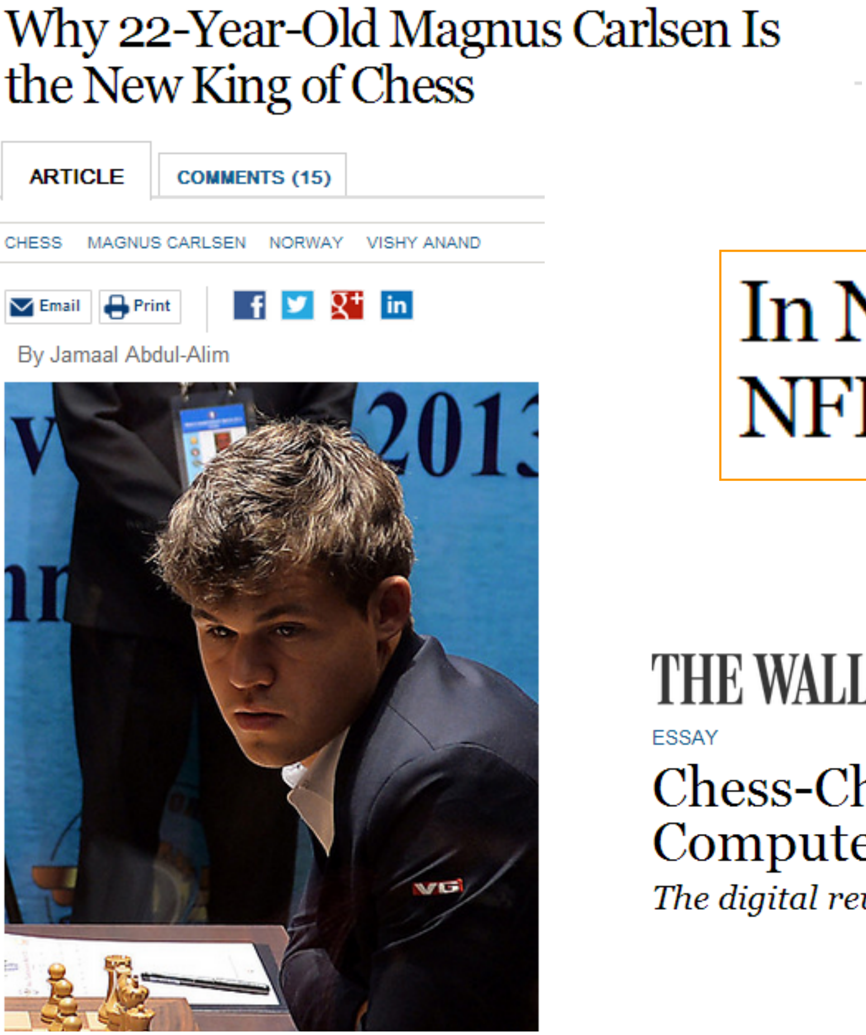
Deep Blue scored its 3 1/2 point to 2 1/2 point triumph in an astonishing 88-minutes. Kasparov shocked the chess world by resigning after only 19 moves with the black pieces.



Championship

The World Chess Championship Match 2016, held from 11 to 30 November, will be contested by 25 year old reigning champion Magnus Carlsen of Norway and his challenger, 26 year old Sergey Karjakin of Russia — and this is the first time that two players who have come of age in the computer era are fighting for the title and represent a generational shift in chess.

But, in practice...



Why 22-Year-Old Magnus Carlsen Is the New King of Chess

By Jamaal Abdul-Alim

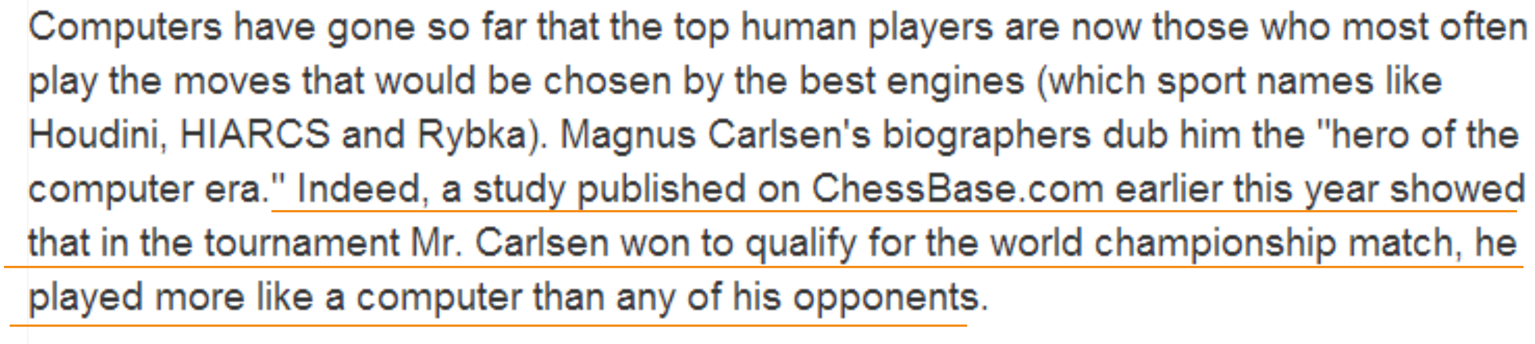
In Norway, Chess Broadcast Spurs NFL-Like Fan Frenzy

THE WALL STREET JOURNAL.

ESSAY

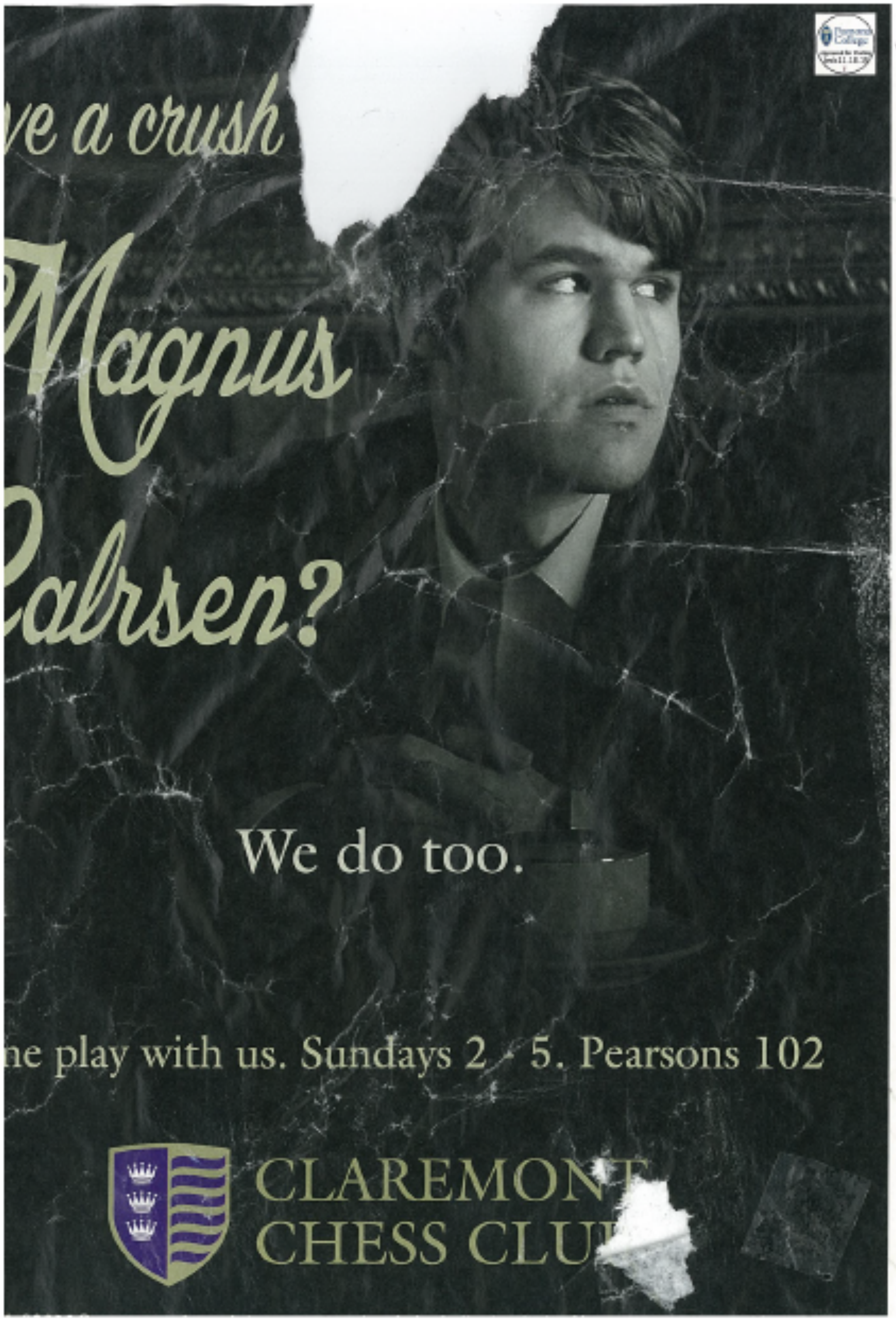
Chess-Championship Results Show Powerful Role of Computers

The digital revolution has pushed human abilities to new heights



Computers have gone so far that the top human players are now those who most often play the moves that would be chosen by the best engines (which sport names like Houdini, HIARCS and Rybka). Magnus Carlsen's biographers dub him the "hero of the computer era." Indeed, a study published on ChessBase.com earlier this year showed that in the tournament Mr. Carlsen won to qualify for the world championship match, he played more like a computer than any of his opponents.

Humans adapt!



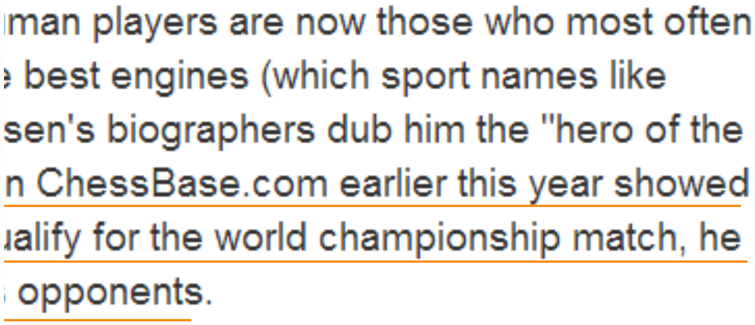
still popular!

vay, Chess Broadcast Spurs
ke Fan Frenzy

'T JOURNAL.

onship Results Show Powerful Role of

as pushed human abilities to new heights

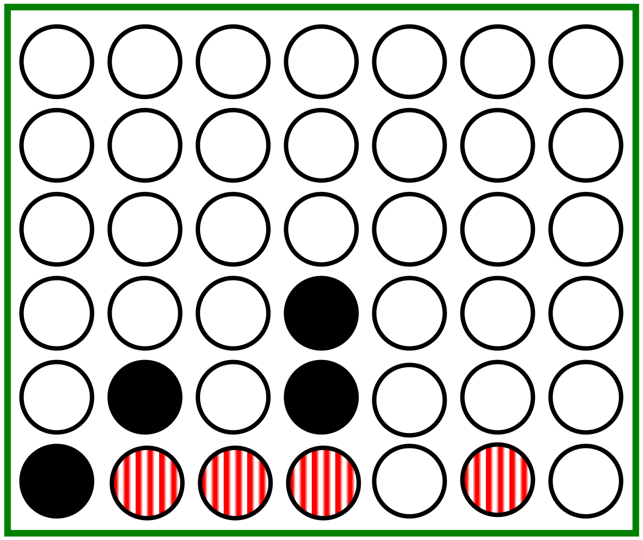


man players are now those who most often
e best engines (which sport names like
sen's biographers dub him the "hero of the
n ChessBase.com earlier this year showed
ualify for the world championship match, he
opponents.

Humans adapt!

Connect 4, Part 2

hw11pr2.py



```
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | |X| | | |
| |X| |X| | | |
|X|O|O|O| |O| |
---------------
 0 1 2 3 4 5 6
```

1 ply + 2 ply

what methods will help?

colsToWin(self, ox)

```
b.colsToWin('O')

b.colsToWin('X')
```

intuition-based tiebreaking

what methods will help?

aiMove(self, ox)

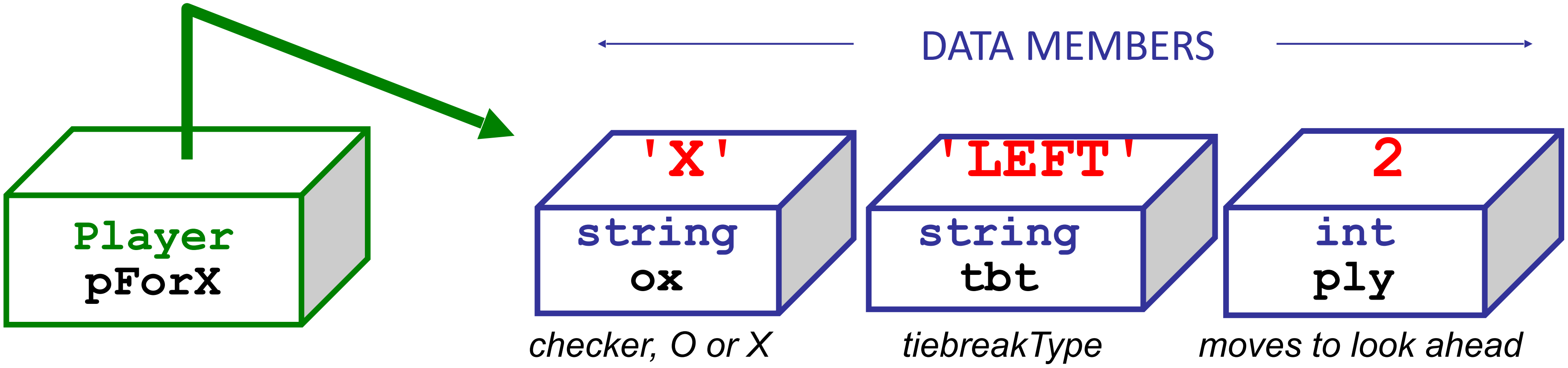
```
b.aiMove('O')

b.aiMove('X')
```

hostGame(self)

The `Player` class

What **data** does a computer AI player need?



Player
pForX

DATA MEMBERS

'X'
string
ox
checker, O or X

'LEFT'
string
tbt
tiebreakType

2
int
ply
moves to look ahead

ox? tbt? ply?

```
x = Player('X', 'LEFT', 42)
x0rn
o0rn
b.playGame( x0rn, o0rn )
```

... perhaps *surprisingly, not so much.*

Looking further ahead... !

How could we write a 3-ply lookahead?
What about 4-ply? ***N-ply?***

How many ply of lookahead would we need to play a ***perfect*** game of Connect Four?

Player's algorithms...

Board

__init__(self, width, height)

allowsMove(self, col)

addMove(self, col, ox)

delMove(self, col)

__repr__(self)

isFull(self)

winsFor(self, ox)

hostGame(self)

playGame(self, pForX, pForO)

Player

__init__(self, ox, tbt, ply)

__repr__(self)

oppCh(self)

scoreBoard(self, b)

scoresFor(self, b) ☆

tiebreakMove(self, scores)

nextMove(self, b)

Why AI is challenging:

Make no mistake about it: computers process numbers - not symbols.

Computers can only help us to the extent that we can ***arithmetize*** an activity.

- paraphrasing Alan Perlis

`scoreBoard(self,b)`

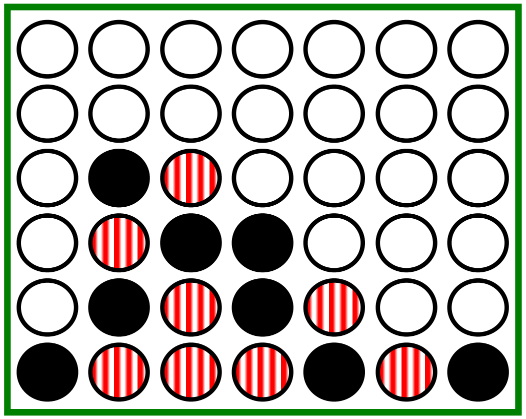
● `'X'`

◍ `'O'`

Returns a *score* for any board, **`b`**

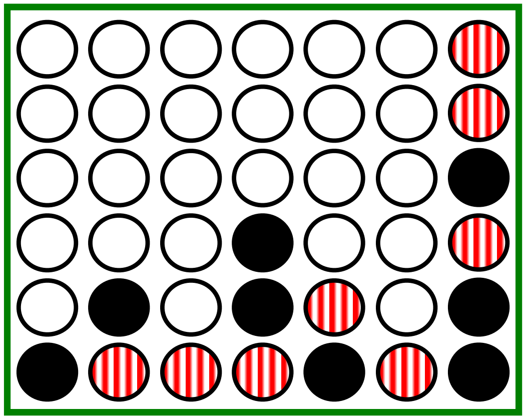
A simple system:

100.0	**50.0**	**0.0**
for a win	**for anything else**	**for a loss**



Score for ●

Score for ◍



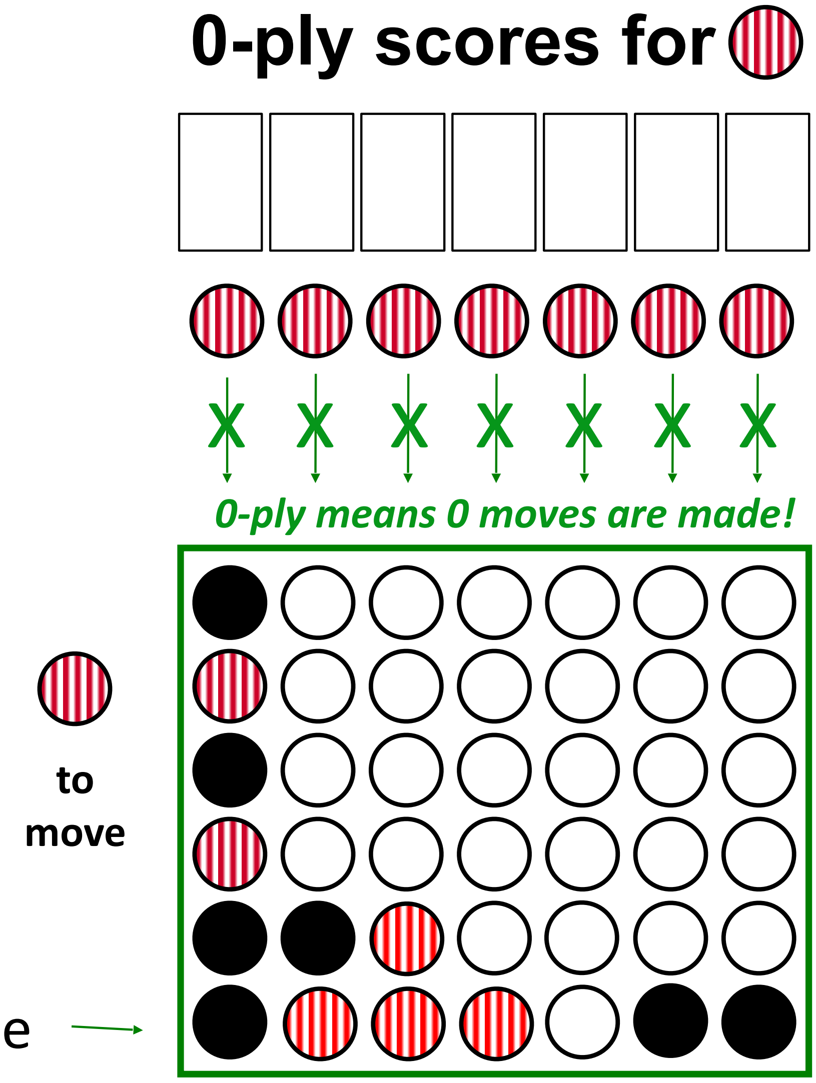
Score for ●

Score for ◍

`scoresFor` at 0 ply...

What should **`scoresFor`** return for ◍ with **ply == 0**

0 ply is a Zen-like approach: ***exist only in the present***



●	○	○	○	○	○	○
◍	○	○	○	○	○	○
●	○	○	○	○	○	○
◍	○	○	○	○	○	○
●	●	◍	○	○	○	○
●	◍	◍	◍	○	●	●

`o0.scoresFor( b2 )`

We still use **`-1`** as the score into a full column.

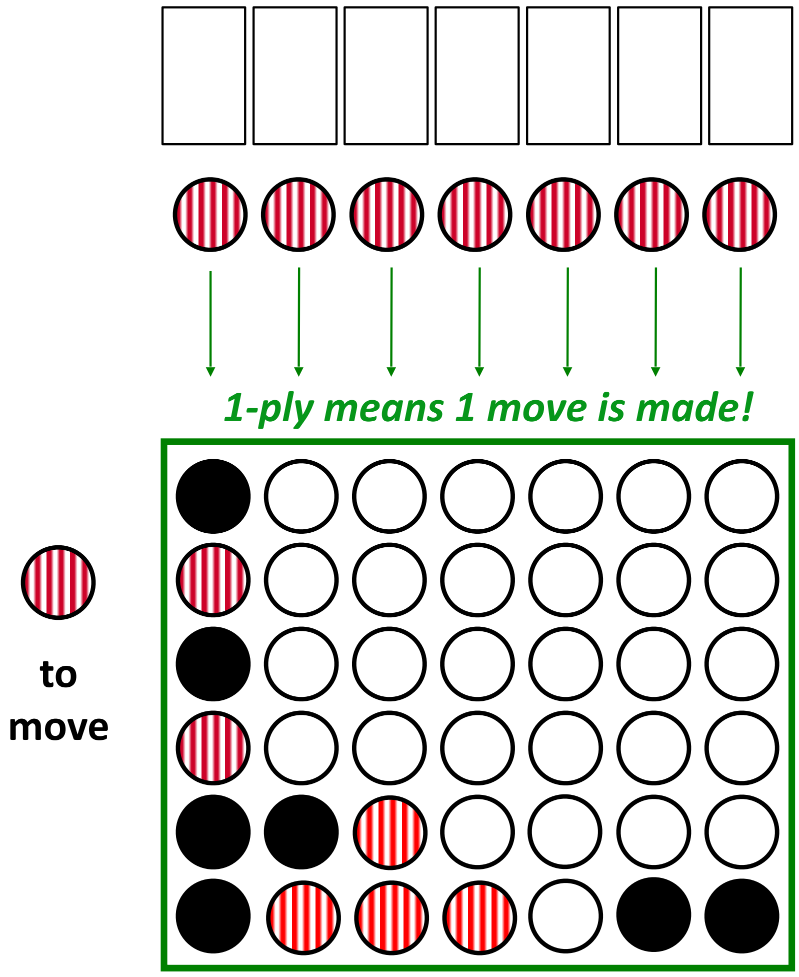
`scoresFor` at 1 ply...

What should **`scoresFor`** return for ◍ with **ply == 1**

A 1-ply lookahead player will "see" an impending victory.

"Gotcha!"

1-ply scores for ◍



1-ply means 1 move is made!

◍ to move

●	○	○	○	○	○	○
◍	○	○	○	○	○	○
●	○	○	○	○	○	○
◍	○	○	○	○	○	○
●	●	◍	○	○	○	○
●	◍	◍	◍	○	●	●

`o1.scoresFor( b2 )`

I try 1 ply!

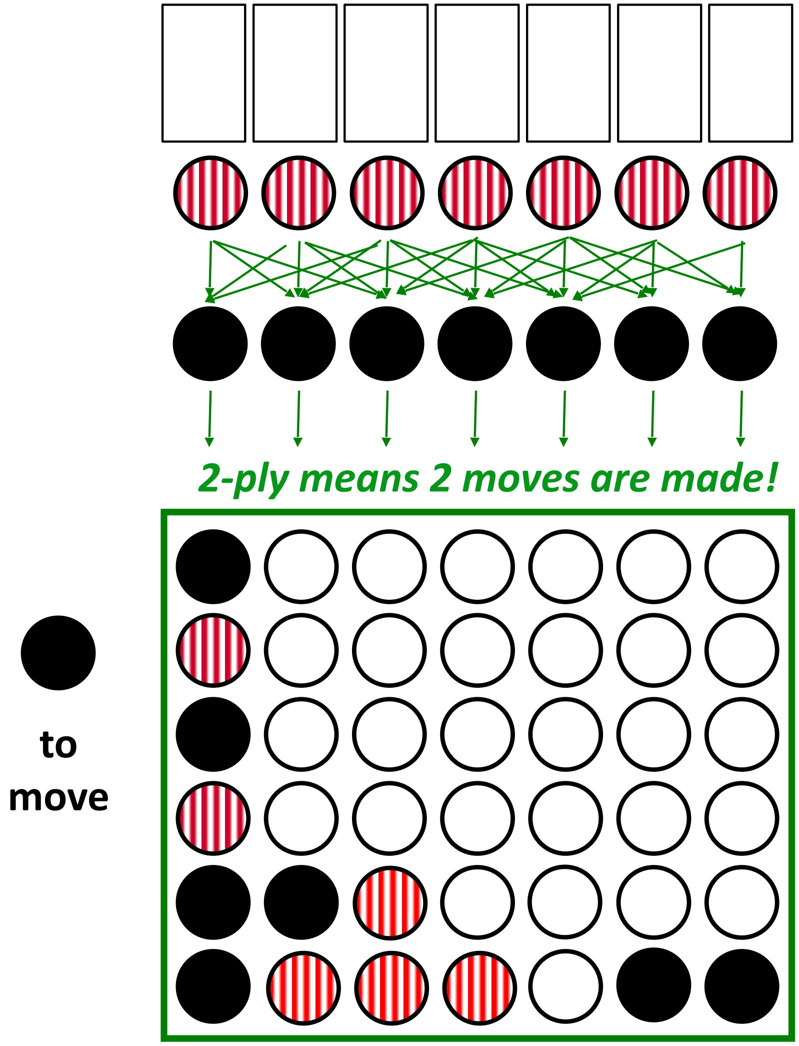
`scoresFor` at 2 ply for ●

What should **`scoresFor`** return for ● with **ply == 2**

A 2-ply lookahead player will see a way to win or block the opponent's win

"Gotcha!" + "Uh Oh..."

2-ply scores for ●



2-ply means 2 moves are made!

● to move

`x2.scoresFor( b2 )`

What about 3-ply?

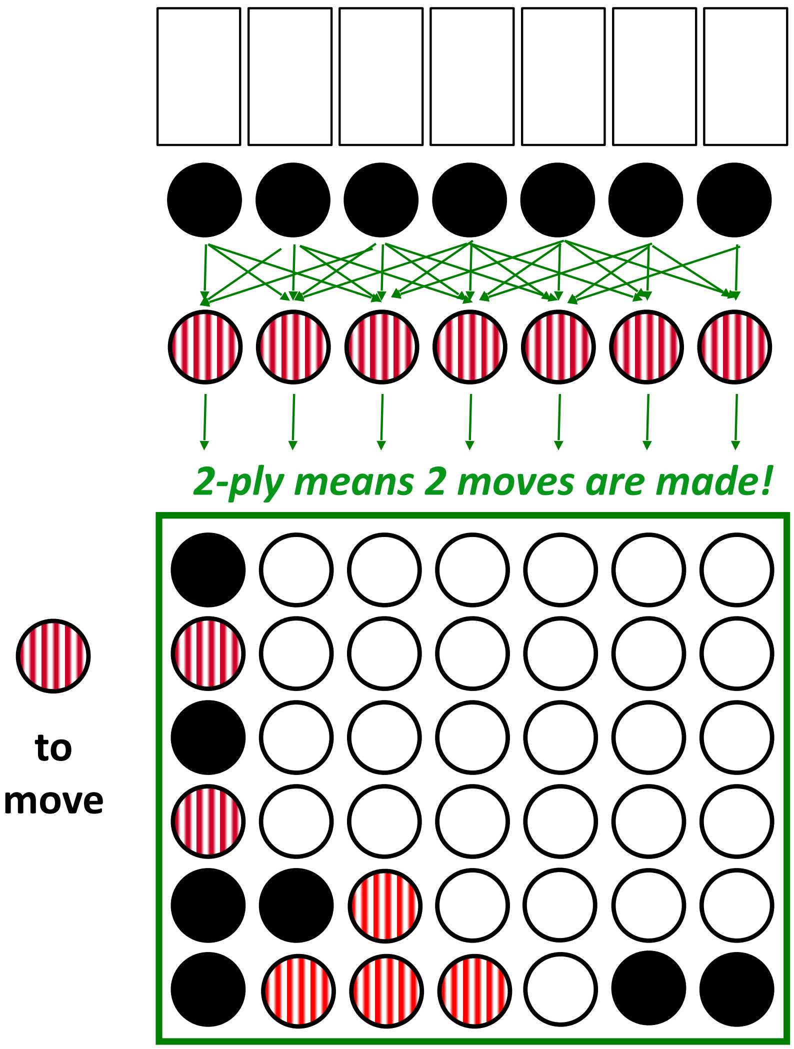
`scoresFor` at 2 ply for 🔴

What should **`scoresFor`** return for 🔴 with **ply == 2**

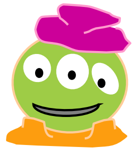
A 2-ply lookahead player will see a way to win or block the opponent's win

"Gotcha!" + "Uh Oh..."

2-ply scores for 🔴



```
o2.scoresFor( b2 )
```



What about 3-ply?

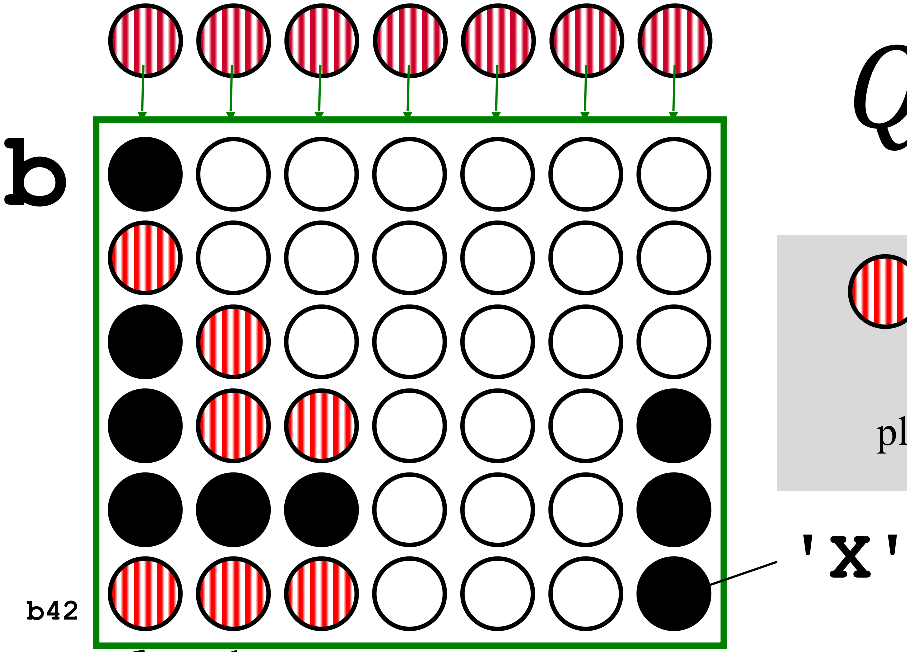
So many ply!

Quiz

Fill in the list of scores returned by **`scoresFor`**

The same move is evaluated at each ply... it's just evaluated farther into the future!

Each row is different in at least 1 score...



'O'

you are playing 'O'

	col 0	col 1	col 2	col 3	col 4	col 5	col 6
scoresFor(b) ox == 'O' and **ply == 0**							
scoresFor(b) ox == 'O' and **ply == 1**							
scoresFor(b) ox == 'O' and **ply == 2**							
scoresFor(b) ox == 'O' and **ply == 3**							

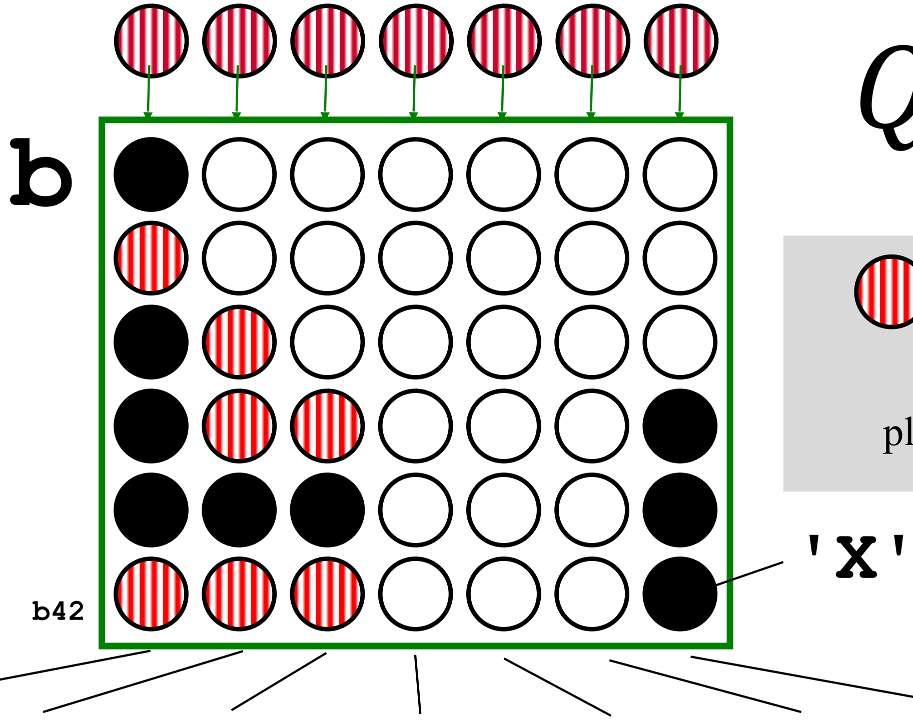
So many ply!

Quiz

Fill in the list of scores returned by **`scoresFor`**

The same move is evaluated at each ply... it's just evaluated farther into the future!

Each row is different in at least 1 score...



▨ 'O'

you are playing 'O'

	col 0	col 1	col 2	col 3	col 4	col 5	col 6
`scoresFor(b)` ox == 'O' and **ply == 0**	-1	50	50	50	50	50	50
`scoresFor(b)` ox == 'O' and **ply == 1**	-1						
`scoresFor(b)` ox == 'O' and **ply == 2**	-1						
`scoresFor(b)` ox == 'O' and **ply == 3**	-1						

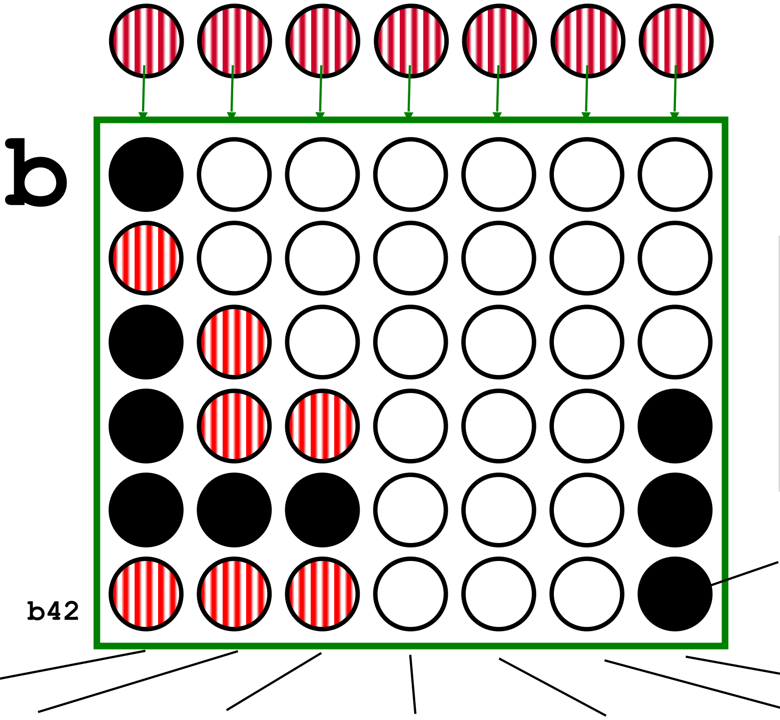
So many ply!

Quiz

Fill in the list of scores returned by **`scoresFor`**

The same move is evaluated at each ply... it's just evaluated farther into the future!

Each row is different in at least 1 score...



● 'O'

you are playing `'O'`

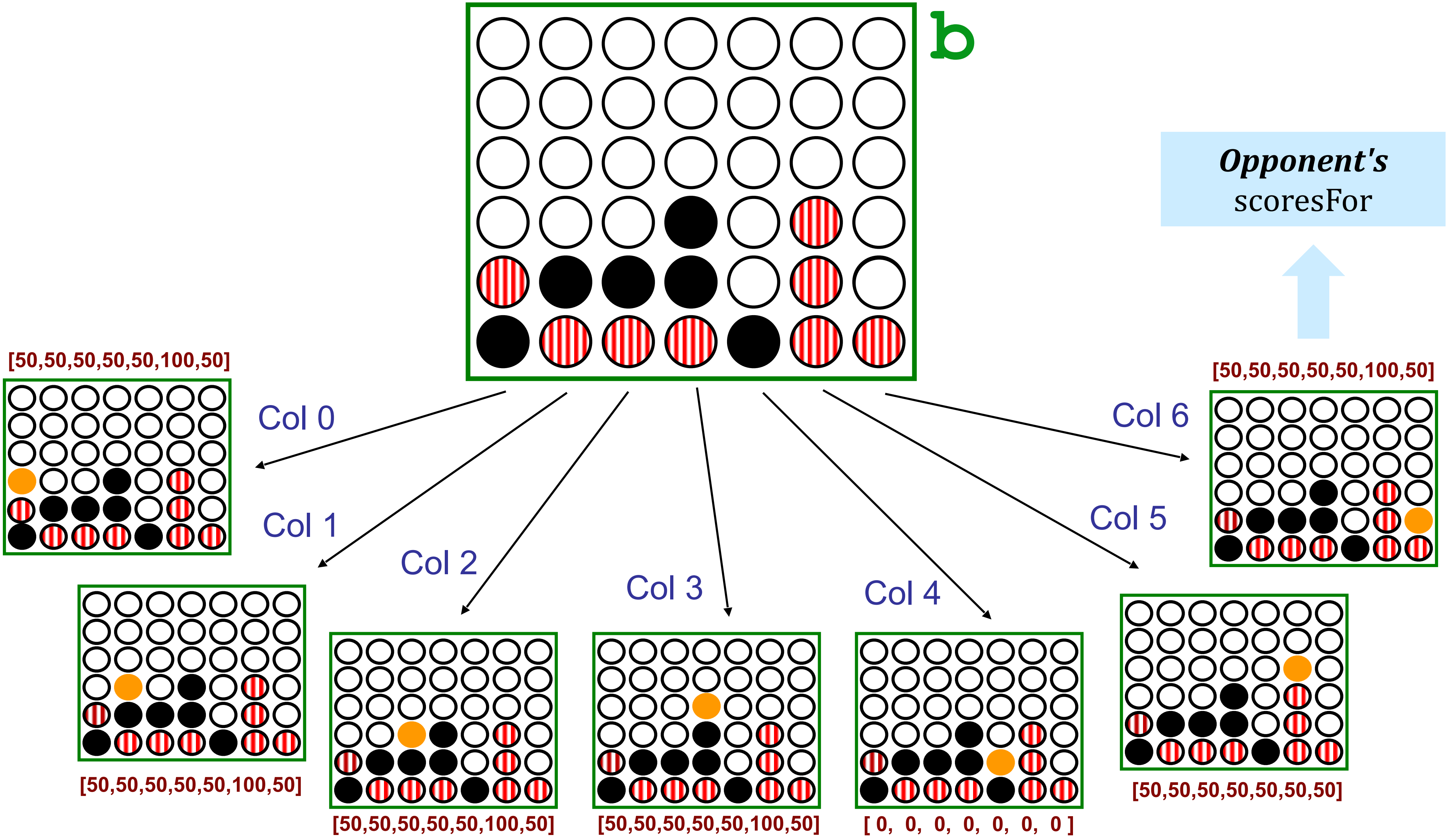
	col 0	col 1	col 2	col 3	col 4	col 5	col 6
`scoresFor(b)` ox == 'O' and **ply == 0**	-1	50	50	50	50	50	50
`scoresFor(b)` ox == 'O' and **ply == 1**	-1	50	50	100	50	50	50
`scoresFor(b)` ox == 'O' and **ply == 2**	-1	0	0	100	0	0	50
`scoresFor(b)` ox == 'O' and **ply == 3**	-1	0	0	100	0	0	

?

Idea: scoresFor

(self) 'X' ●

new 'X' 🟠



b

Col 0

[50,50,50,50,50,100,50]

Col 1

[50,50,50,50,50,100,50]

Col 2

[50,50,50,50,50,100,50]

Col 3

[50,50,50,50,50,100,50]

Col 4

[0, 0, 0, 0, 0, 0, 0]

Col 5

[50,50,50,50,50,50,50]

Col 6

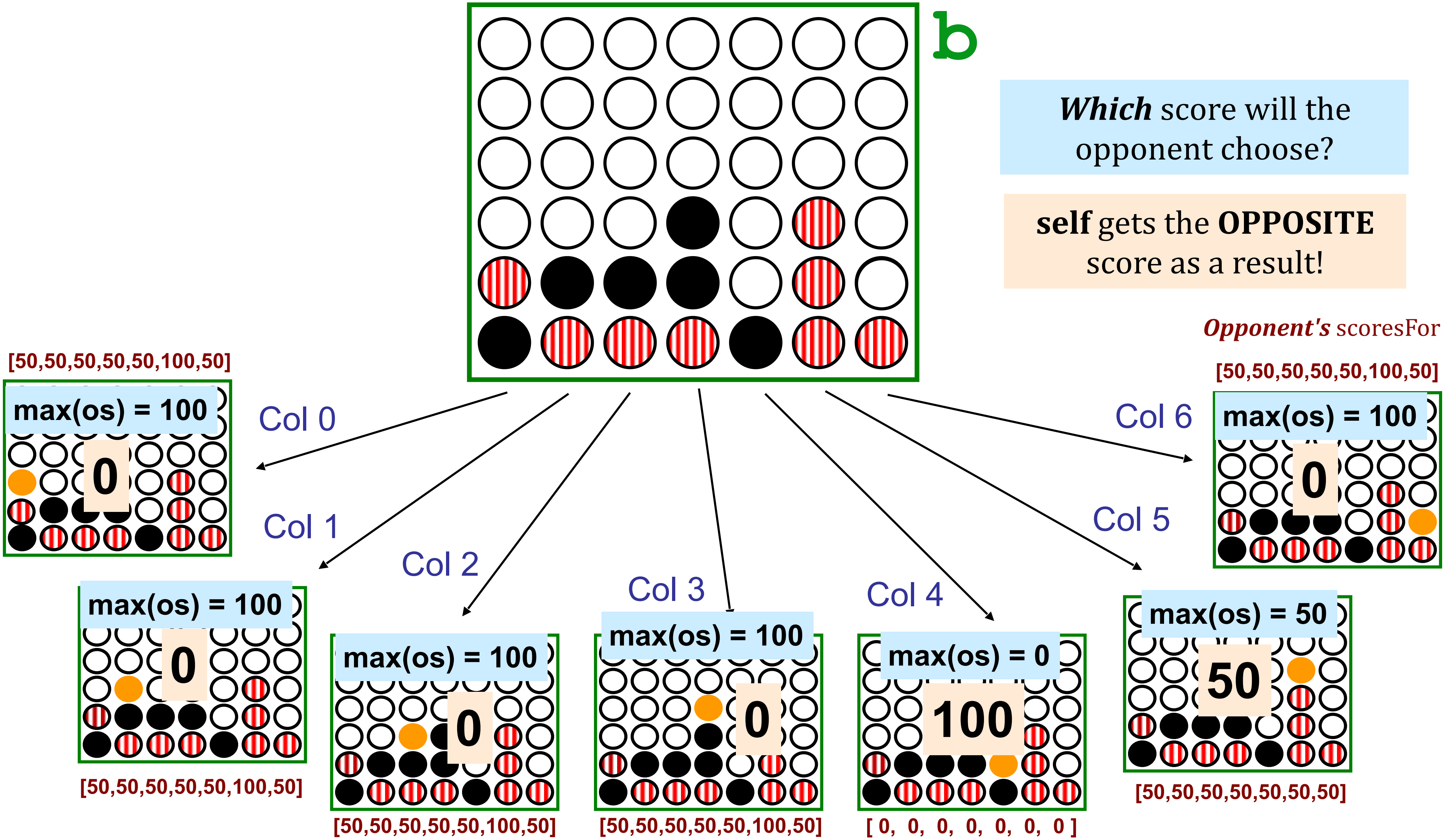
[50,50,50,50,50,100,50]

Opponent's scoresFor

Idea: `scoresFor`

`(self) 'X'` ●

`new 'X'` 🟠



Which score will the opponent choose?

self gets the **OPPOSITE** score as a result!

Opponent's scoresFor

scoresFor

(self) 'X' ●

new 'X' ●

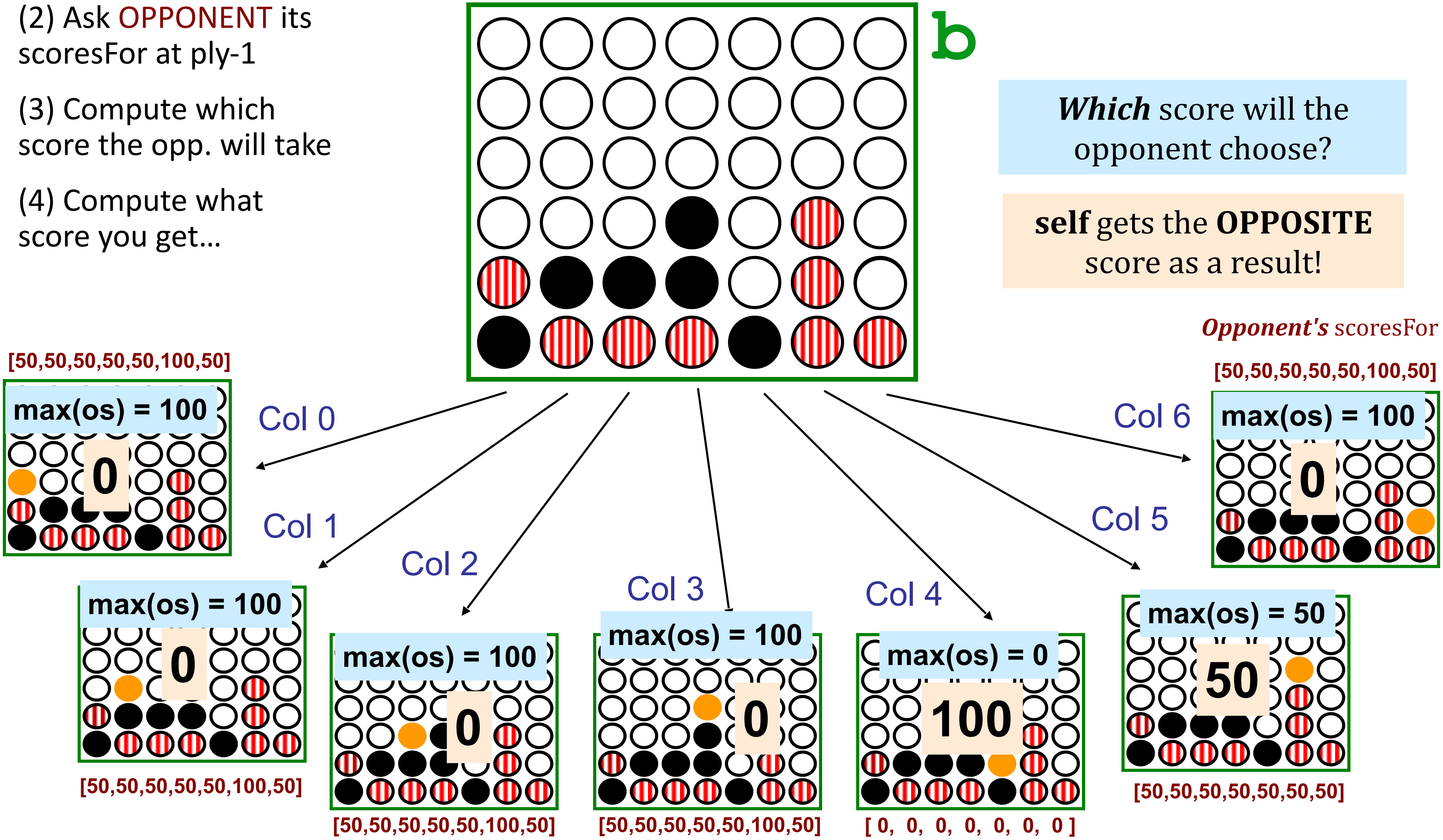
(0) Suppose you're playing at 2 ply...

(1) Make ALL moves!

(2) Ask OPPONENT its scoresFor at ply-1

(3) Compute which score the opp. will take

(4) Compute what score you get...



Which score will the opponent choose?

self gets the **OPPOSITE** score as a result!

?

Strategic thinking == intelligence

Two-player games have been a key focus of AI as long as computers have been around…

Alan Turing memorial
Manchester, England

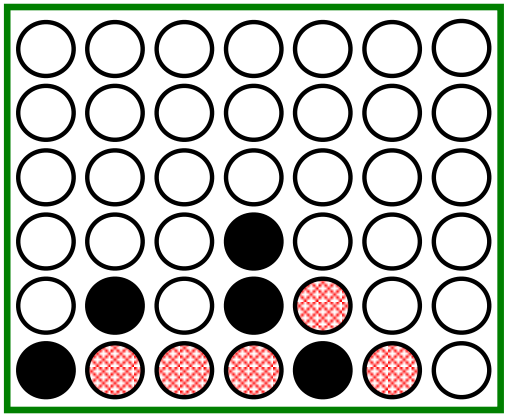
In **1945**, Alan Turing predicted that computers would be better chess players than people in ~ 50 years…

and thus would have achieved intelligence.

Strategic thinking != intelligence

computers

good at looking to find winning combinations of moves



humans

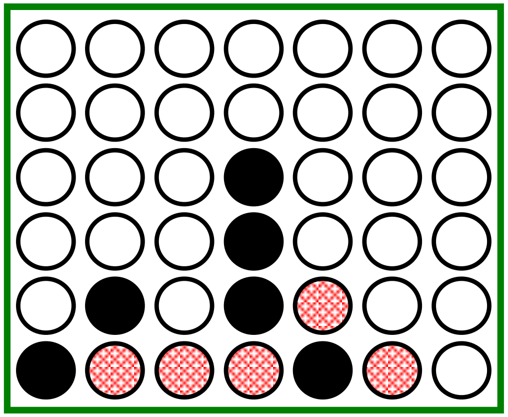
good at evaluating the strength of a board for a player

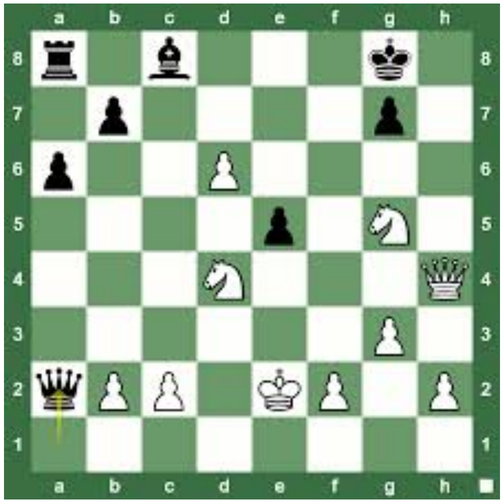
... humans and computers have different relative strengths in these games.

Humans play via "look-up table"

An experiment (by A. deGroot) was performed in which chess positions were shown to novice and expert players for a few seconds...

- experts could reconstruct these perfectly
- novice players did far worse...



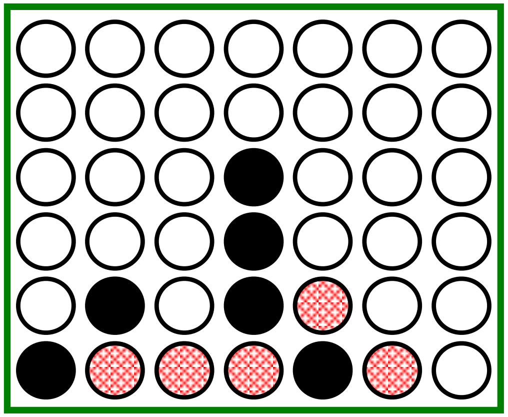


Adriaan de Groot

Humans play via "look-up"

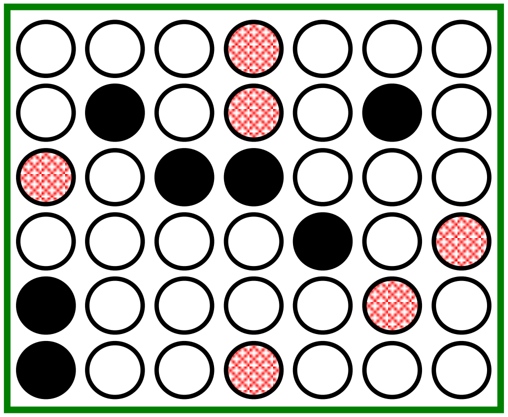
An experiment (by A. deGroot) was performed in which chess positions were shown to novice and expert players for a few seconds...

- experts could reconstruct these perfectly
- novice players did far worse...



Random chess positions (not legal ones) were then shown to the two groups

- experts and novices did **equally badly** at reconstructing them!



Adriaan de Groot

Connecting Connect Four …

Connect 4

How complex are these games? *Least? Most?*

… to other strategy games.

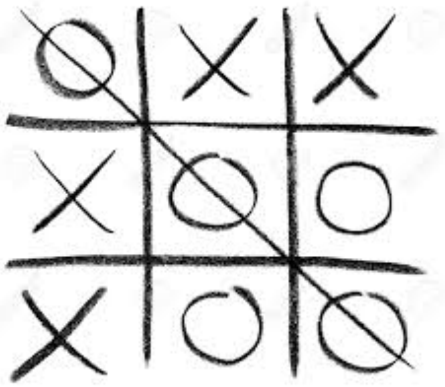
Connecting Connect Four ...

checkers

chess

Connect 4

reversi



tic-tac-toe

Go

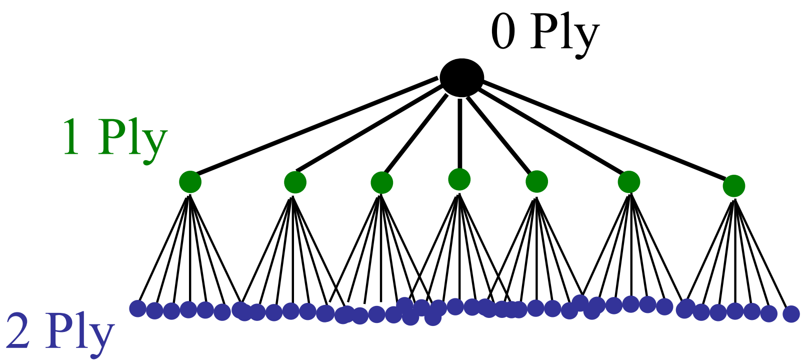
How complex are these games?
Least? Most?

... to other strategy games.

Games' Branching Factors

On average, Connect 4 players have **seven choices** per move.

Chess players have more, **perhaps around 40**, possible choices in a given move.



Boundaries for ***qualitatively*** different games…

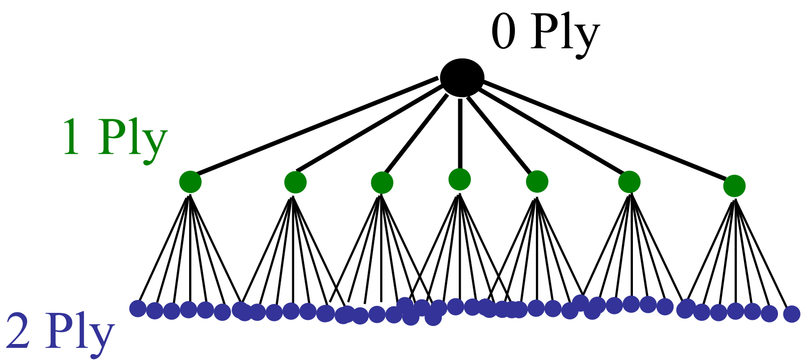
Branching Factors
for different two-player games

"solved" games	Tic-tac-toe	4
	Connect Four	7
	Checkers	10
computer-dominated	Reversi	30
	Chess	40
human-dominated	Go	300
	Arimaa	17,000

Games' Branching Factors

On average, Connect 4 players have **seven choices** per move.

Chess players have more, **perhaps around 40**, possible choices in a given move.

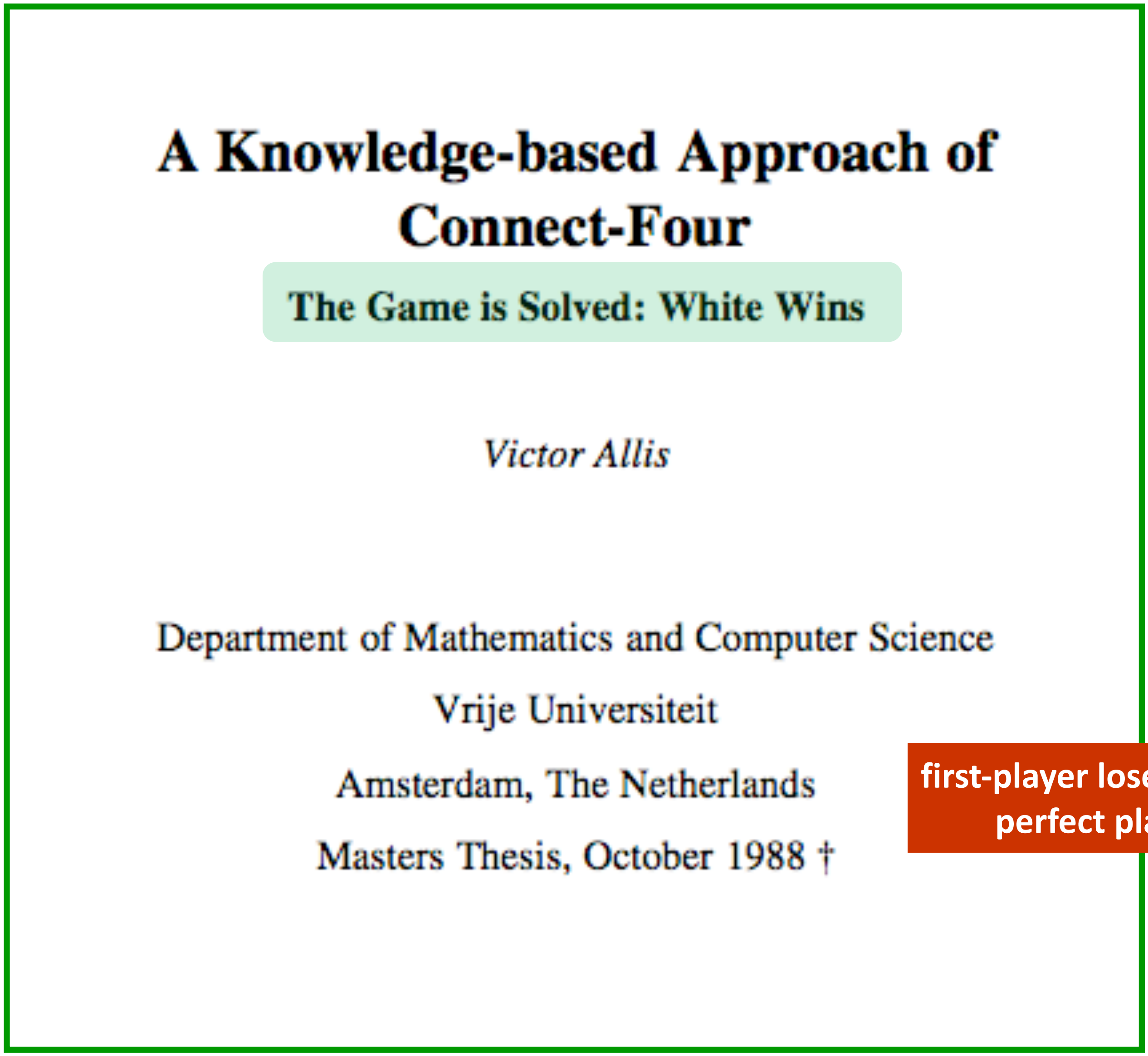


Boundaries for ***qualitatively*** different games…

Branching Factors
for different two-player games

“solved” games	Tic-tac-toe	4
	Connect Four	7
	Checkers	10
computer-dominated	Reversi	30
	Chess	40
human-dominated	Go	300
	Arimaa	17,000

only until 2016 → human-dominated



A Knowledge-based Approach of Connect-Four

The Game is Solved: White Wins

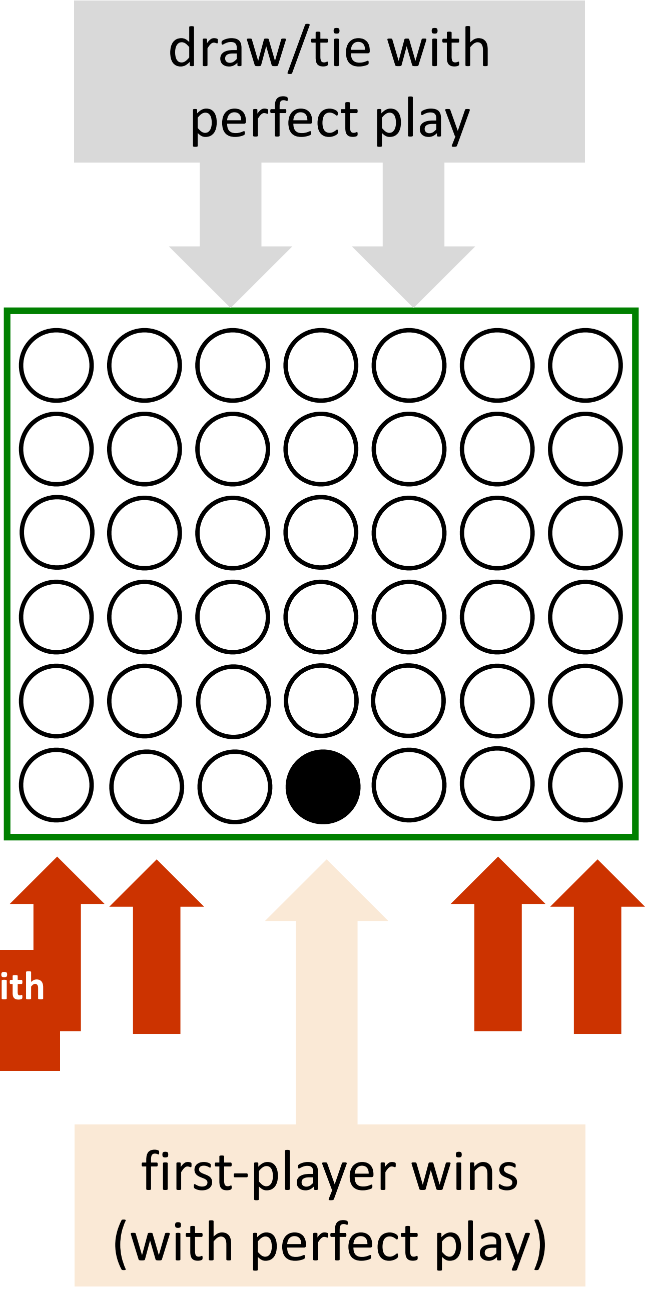
Victor Allis

Department of Mathematics and Computer Science

Vrije Universiteit

Amsterdam, The Netherlands

Masters Thesis, October 1988 †

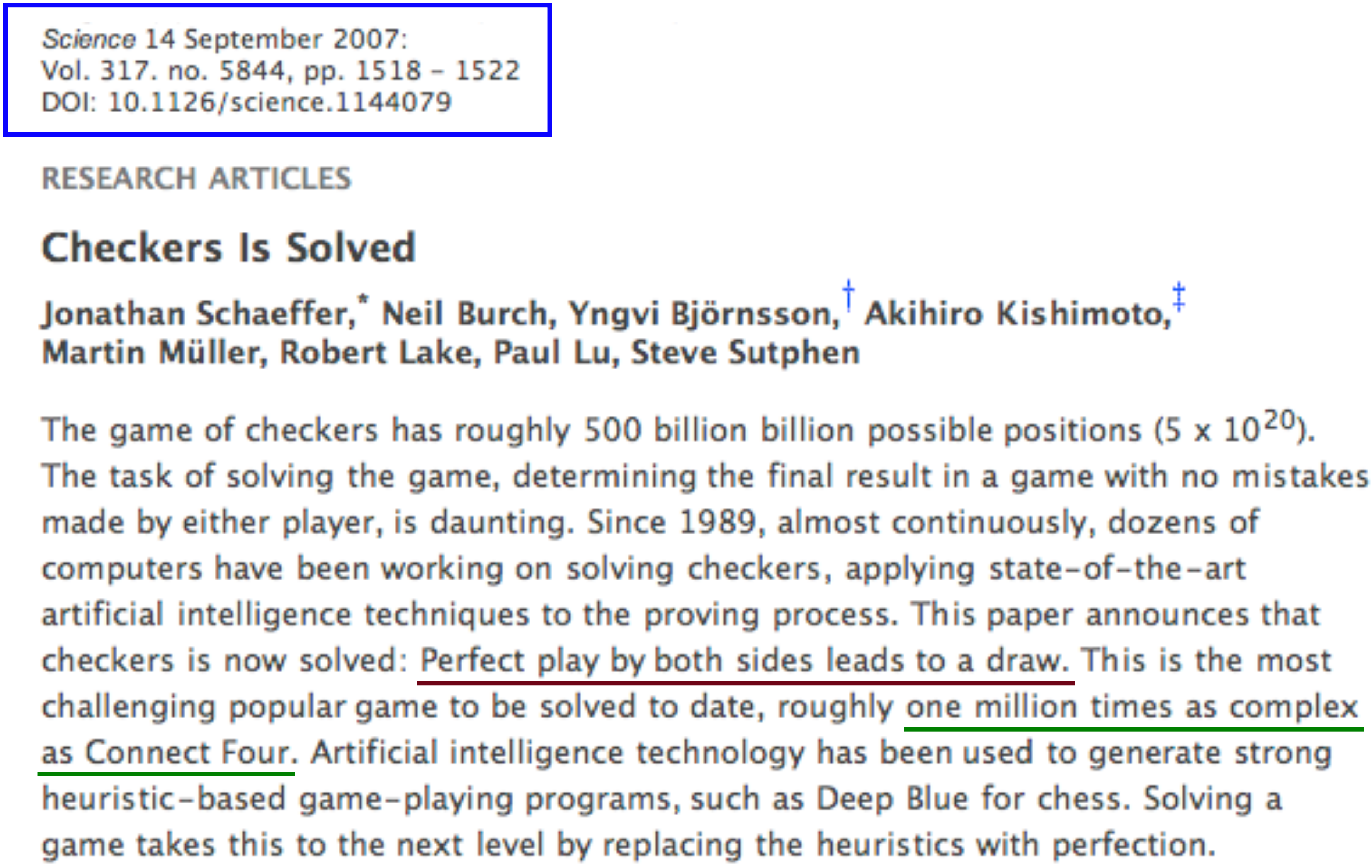


draw/tie with perfect play

first-player loses (with perfect play)

first-player wins (with perfect play)

Connect 4 was solved in **1988**.



Science 14 September 2007:
Vol. 317. no. 5844, pp. 1518 - 1522
DOI: 10.1126/science.1144079

RESEARCH ARTICLES

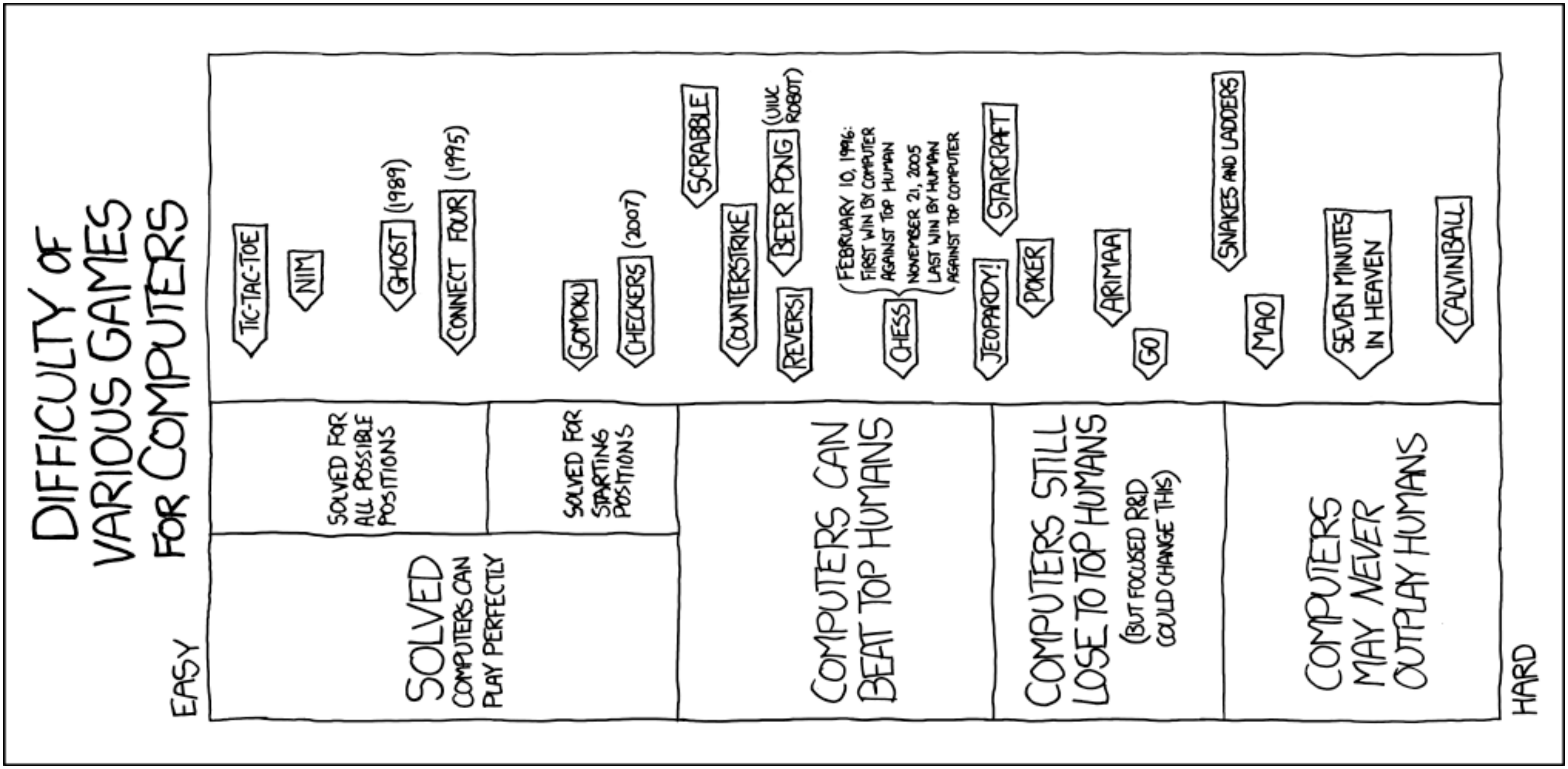
Checkers Is Solved

Jonathan Schaeffer,[*] Neil Burch, Yngvi Björnsson,[†] Akihiro Kishimoto,[‡] Martin Müller, Robert Lake, Paul Lu, Steve Sutphen

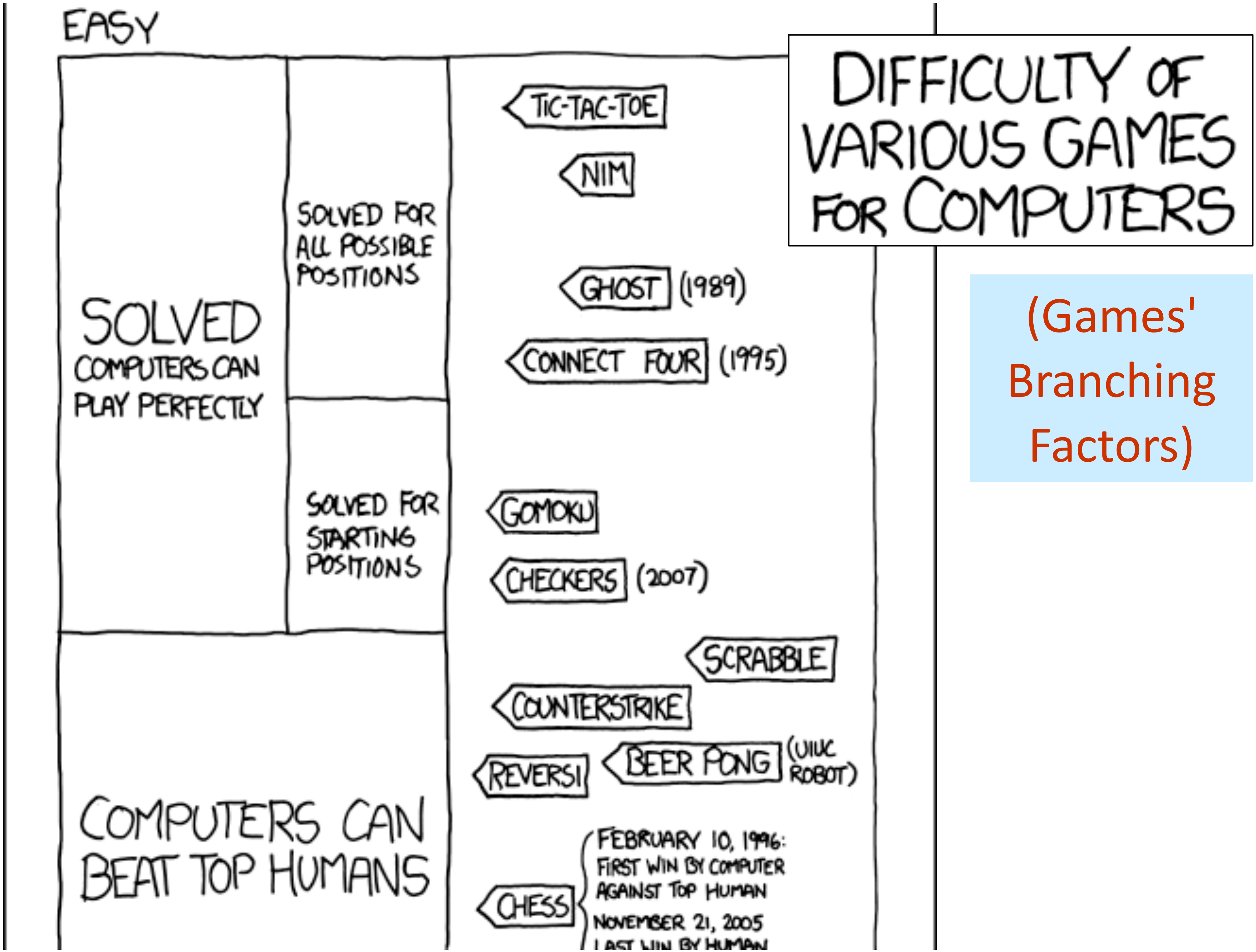
The game of checkers has roughly 500 billion billion possible positions (5×10^{20}). The task of solving the game, determining the final result in a game with no mistakes made by either player, is daunting. Since 1989, almost continuously, dozens of computers have been working on solving checkers, applying state-of-the-art artificial intelligence techniques to the proving process. This paper announces that checkers is now solved: Perfect play by both sides leads to a draw. This is the most challenging popular game to be solved to date, roughly one million times as complex as Connect Four. Artificial intelligence technology has been used to generate strong heuristic-based game-playing programs, such as Deep Blue for chess. Solving a game takes this to the next level by replacing the heuristics with perfection.

Checkers was solved in **2007**.

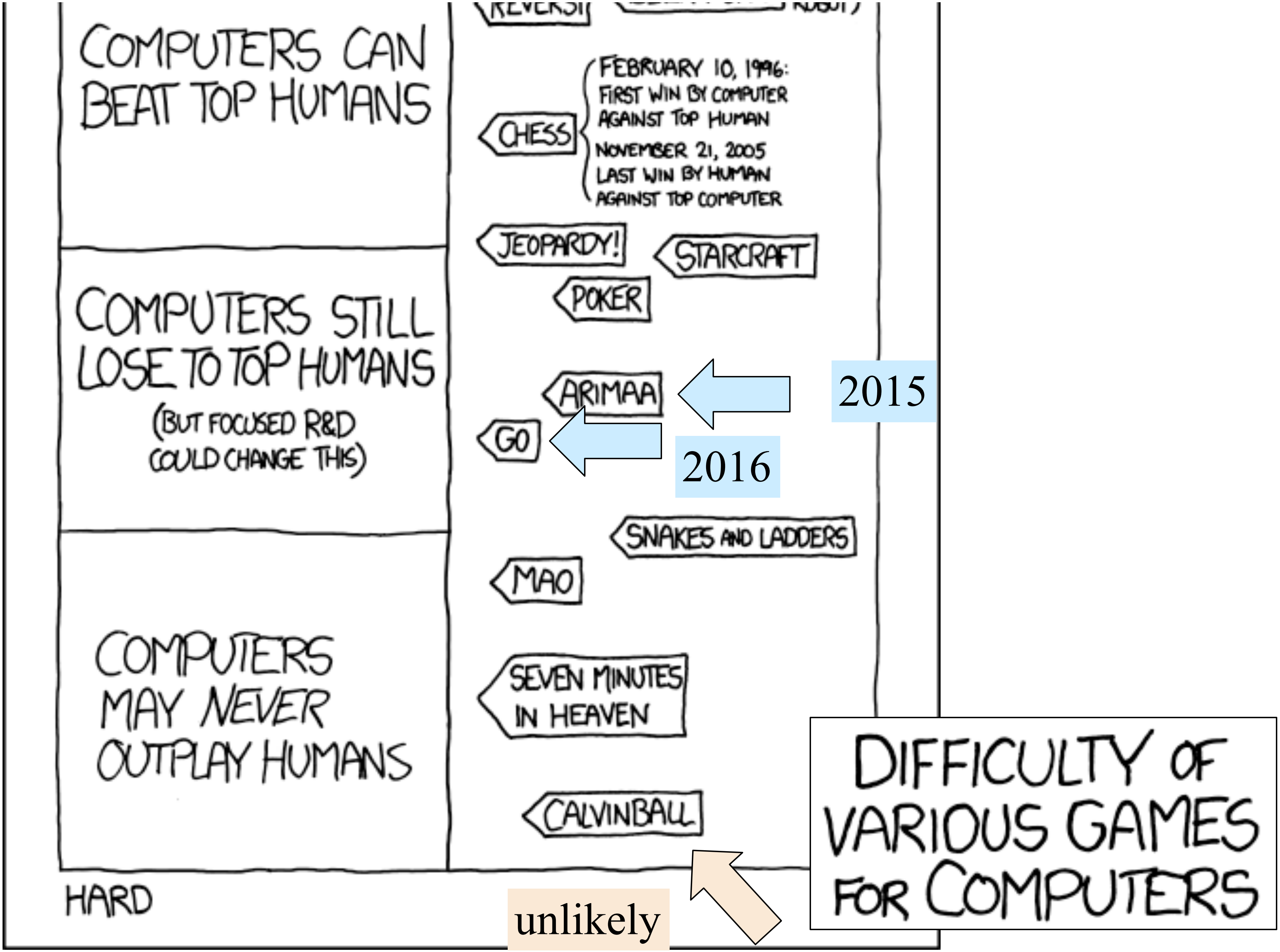
Games' complexity ~ xkcd



~2012


DIFFICULTY OF VARIOUS GAMES FOR COMPUTERS
EASY
SOLVED
COMPUTERS CAN PLAY PERFECTLY
SOLVED FOR ALL POSSIBLE POSITIONS
SOLVED FOR STARTING POSITIONS
COMPUTERS CAN BEAT TOP HUMANS
TIC-TAC-TOE
NIM
GHOST (1989)
CONNECT FOUR (1995)
GOMOKU
CHECKERS (2007)
SCRABBLE
COUNTERSTRIKE
REVERSI
BEER PONG (UIUC ROBOT)
CHESS
FEBRUARY 10, 1996: FIRST WIN BY COMPUTER AGAINST TOP HUMAN
NOVEMBER 21, 2005 LAST WIN BY HUMAN

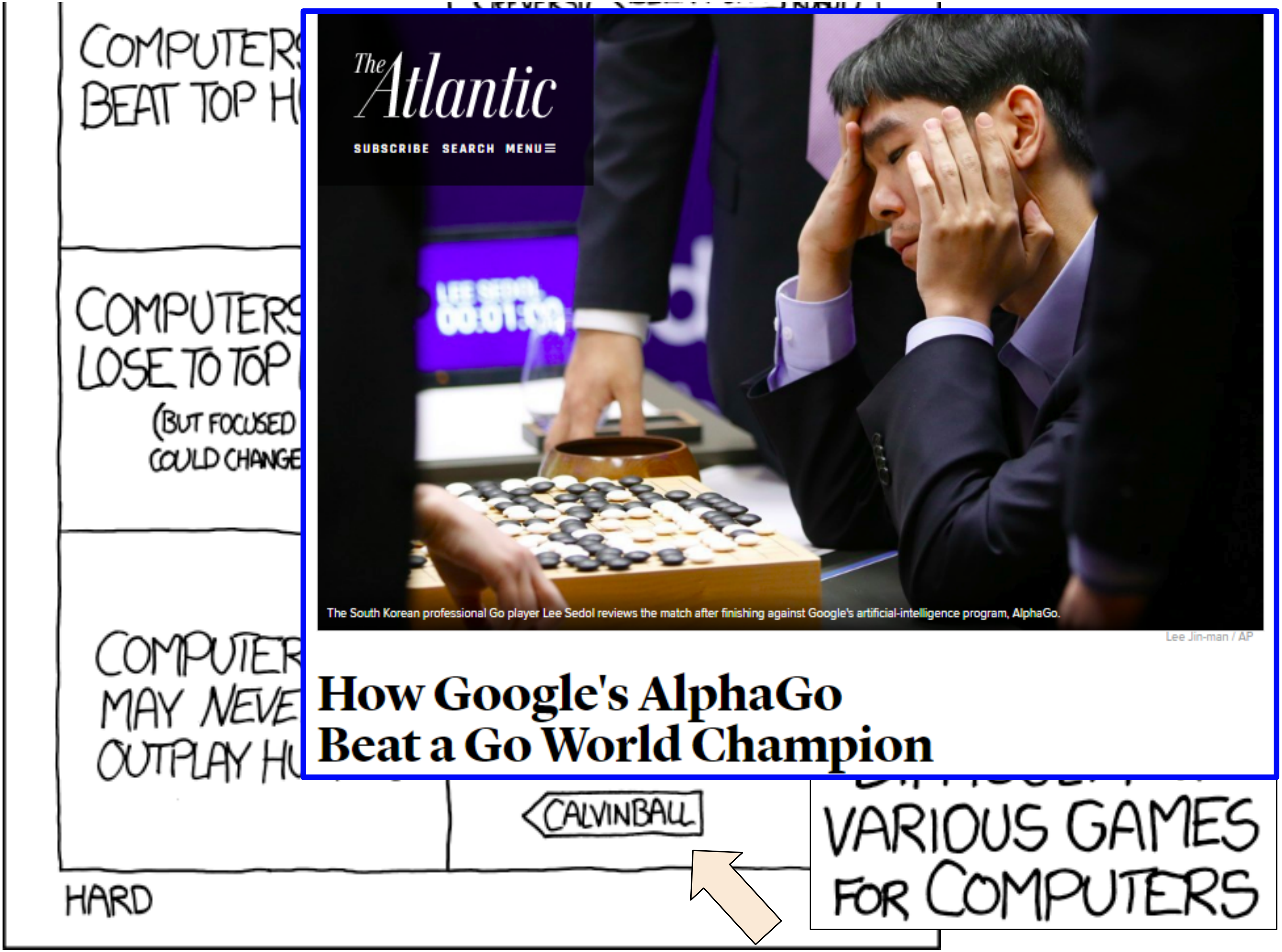
(Games' Branching Factors)


REVERSI
COMPUTERS CAN BEAT TOP HUMANS
CHESS
FEBRUARY 10, 1996: FIRST WIN BY COMPUTER AGAINST TOP HUMAN
NOVEMBER 21, 2005 LAST WIN BY HUMAN AGAINST TOP COMPUTER
JEOPARDY!
STARCRAFT
POKER
COMPUTERS STILL LOSE TO TOP HUMANS
(BUT FOCUSED R&D COULD CHANGE THIS)
ARIMAA
GO
SNAKES AND LADDERS
MAO
COMPUTERS MAY NEVER OUTPLAY HUMANS
SEVEN MINUTES IN HEAVEN
CALVINBALL
HARD
DIFFICULTY OF VARIOUS GAMES FOR COMPUTERS

2015

2016

unlikely



The *Atlantic*

SUBSCRIBE SEARCH MENU ≡

The South Korean professional Go player Lee Sedol reviews the match after finishing against Google's artificial-intelligence program, AlphaGo.

Lee Jin-man / AP

How Google's AlphaGo Beat a Go World Champion

COMPUTERS BEAT TOP H
COMPUTERS LOSE TO TOP
(BUT FOCUSED COULD CHANGE
COMPUTER MAY NEVE OUTPLAY HU
CALVINBALL
HARD
VARIOUS GAMES FOR COMPUTERS

COMPUTERS
The Atlantic
COMPUTERS
MAY NEVE
OUTPLAY HU
CALVINBALL
HARD
VARIOUS GAMES
FOR COMPUTERS
00:
!

nature.com > nature > articles > article

MENU

nature
International journal of science

https://www.youtube.com/watch?v=4Sm922Xp5N4

Altmetric: 2133 Citations: 1 More detail »

Article

Mastering the game of Go without human knowledge

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel & Demis Hassabis

Nature **550**, 354–359 (19 October 2017)
doi:10.1038/nature24270
Download Citation

Received: 07 April 2017
Accepted: 13 September 2017
Published online: 18 October 2017

le's artificial-intelligence program, AlphaGo.

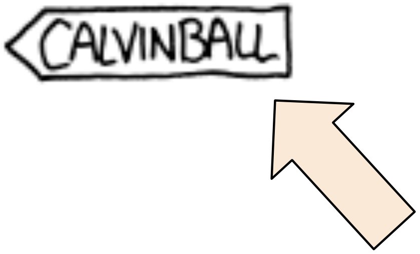
Lee Jin-man / AP

How Google's AlphaGo Beat a Go World Champion

unlikely

IT'S SATURDAY! WHAT DO YOU WANT TO DO?
ANYTHING BUT PLAY AN ORGANIZED SPORT.
WANT TO PLAY CALVINBALL?
YEAH!
NO SPORT IS LESS ORGANIZED THAN CALVINBALL!
NEW RULE! NEW RULE! IF YOU DON'T TOUCH THE 30-YARD BASE WICKET WITH THE FLAG, YOU HAVE TO HOP ON ONE FOOT!
12
4

I DIDN'T SEE YOU TOUCH THE OPPOSITE POLE! YOU HAVE TO DECLARE IT!
I DECLARED IT OPPOSITELY BY NOT DECLARING IT. START SINGING.



DIFFICULTY OF VARIOUS GAMES FOR COMPUTERS