

## AI for Connect Four (cont.)

Computer Science 111  
Boston University  
Vahid Azadeh-Ranjbar, Ph.D.

### Inheritance in PS 9

- `Player` – the superclass
  - includes fields and methods needed by all C4 players
  - in particular, a `next_move` method
  - use this class for human players
- `RandomPlayer` – a subclass for an *unintelligent* computer player
  - no new fields
  - overrides `next_move` with a version that chooses at random from the non-full columns
- `AIPlayer` – a subclass for an "intelligent" computer player
  - uses AI techniques
  - new fields for details of its strategy
  - overrides `next_move` with a version that tries to determine the best move!

## 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

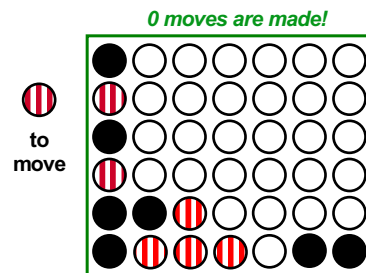
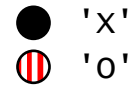
## "Arithmetizing" Connect Four

- Our AIPlayer assigns a score to each possible move
  - i.e., to each column
- It *looks ahead* some number of moves into the future to determine the score.
  - *lookahead* = # of future moves that the player considers
- Scoring columns:
  - 1: an already *full column*
  - 0: if we choose this column, it will result in a *loss* at some point during the player's lookahead
  - 100: if we choose this column, it will result in a *win* at some point during the player's lookahead
  - 50: if we choose this column, it will result in *neither a win nor a loss* during the player's lookahead

## A Lookahead of 0

- A lookahead-0 player only assesses the current board (0 moves!).

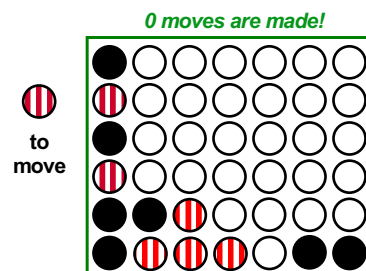
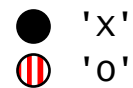
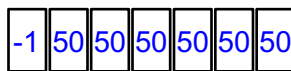
LA-0 scores for 



## A Lookahead of 0

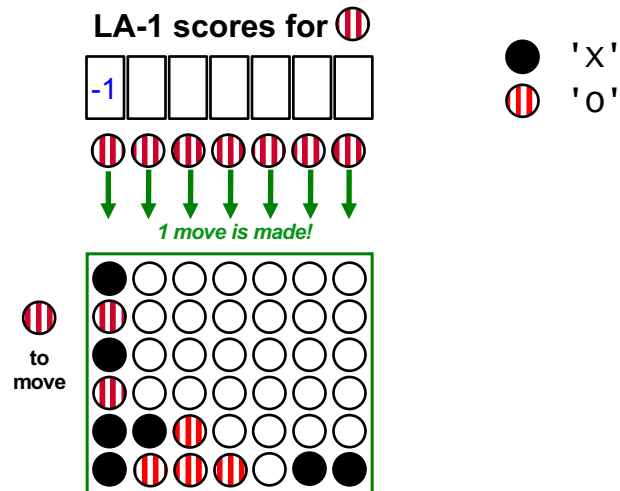
- A lookahead-0 player only assesses the current board (0 moves!).

LA-0 scores for 



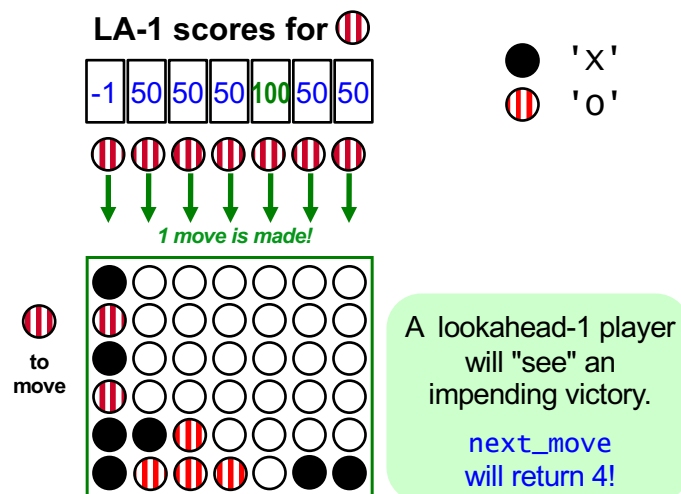
## A Lookahead of 1

- A lookahead-1 player assesses the outcome of *only* the considered move.



## A Lookahead of 1

- A lookahead-1 player assesses the outcome of *only* the considered move.



## A Lookahead of 1

- A lookahead-1 player assesses the outcome of *only* the considered move.

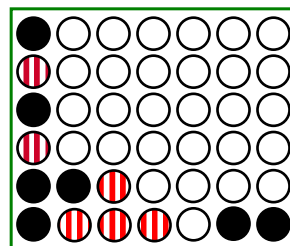
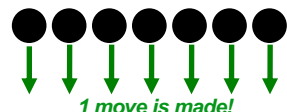
How do these scores change if it is ●'s turn instead of ○'s?

LA-1 scores for ●

-1	50	50	50	100	50	50
----	----	----	----	-----	----	----

● 'X'  
○ 'O'

●  
to  
move



## A Lookahead of 1

- A lookahead-1 player assesses the outcome of *only* the considered move.

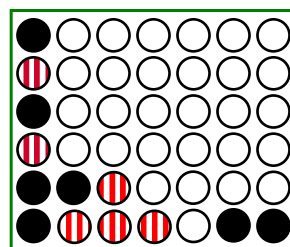
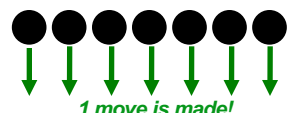
How do these scores change if it is ●'s turn instead of ○'s?

LA-1 scores for ●

-1	50	50	50	50	50	50
----	----	----	----	----	----	----

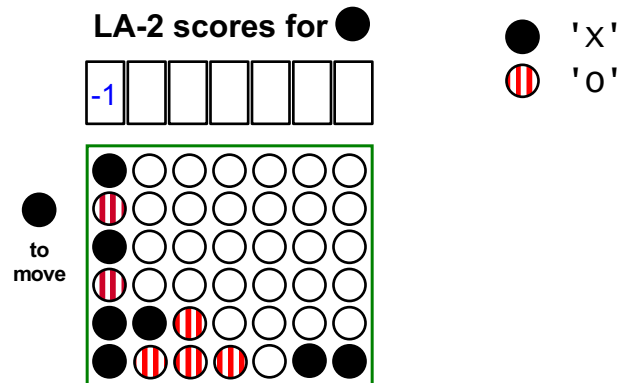
● 'X'  
○ 'O'

●  
to  
move



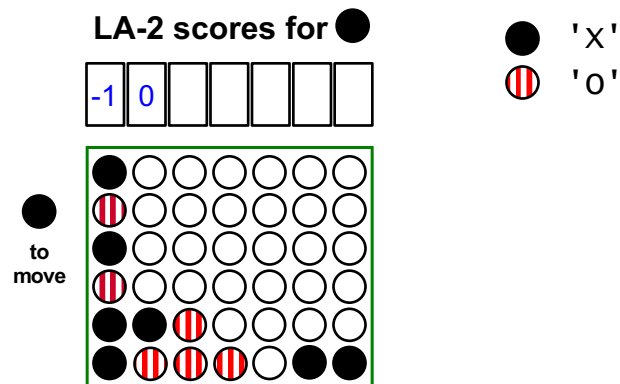
## A Lookahead of 2

- A lookahead-2 player looks 2 moves ahead.
  - what if I make this move, and then my opponent makes *its best move*?
  - **note:** we assume the opponent looks ahead  $2 - 1 = 1$  move



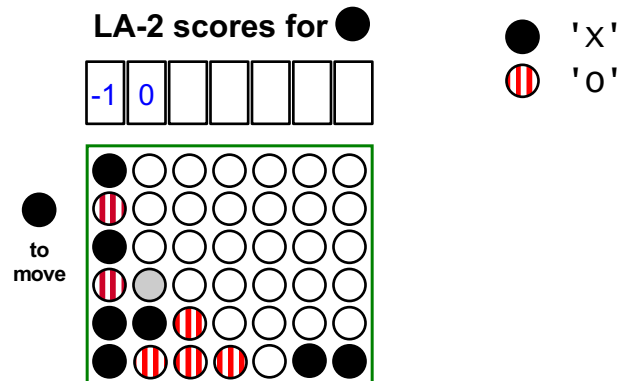
## A Lookahead of 2

- A lookahead-2 player looks 2 moves ahead.
  - what if I make this move, and then my opponent makes *its best move*?
  - **note:** we assume the opponent looks ahead  $2 - 1 = 1$  move



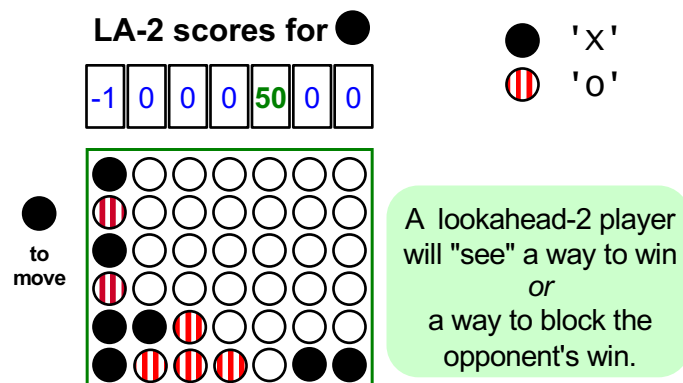
## A Lookahead of 2

- A lookahead-2 player looks 2 moves ahead.
  - what if I make this move, and then my opponent makes *its best move*?
  - **note:** we assume the opponent looks ahead  $2 - 1 = 1$  move



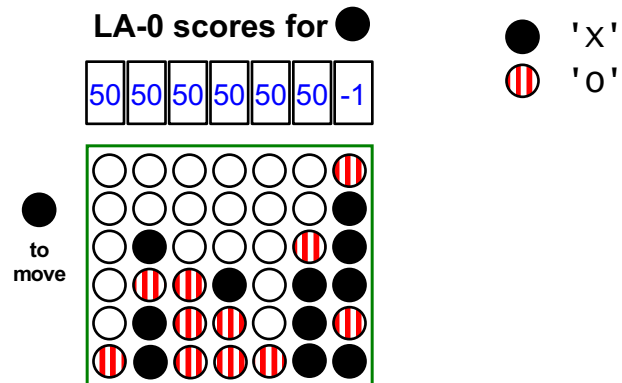
## A Lookahead of 2

- A lookahead-2 player looks 2 moves ahead.
  - what if I make this move, and then my opponent makes *its best move*?
  - **note:** we assume the opponent looks ahead  $2 - 1 = 1$  move



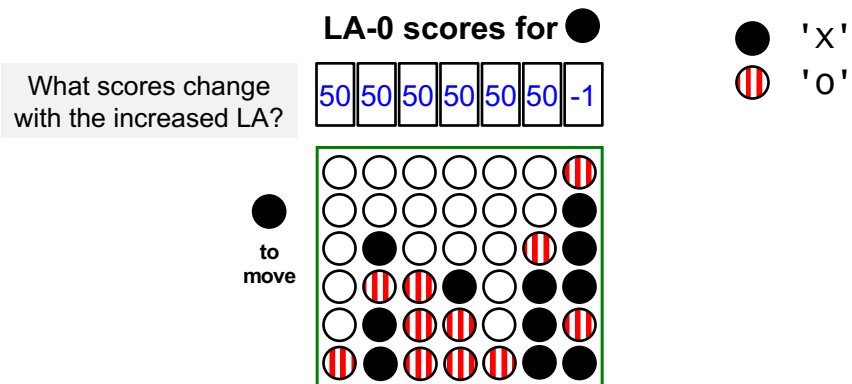
## Example 2: LA-0

- A lookahead-0 player only assesses the current board (0 moves!).



## Example 2: LA-1

- A lookahead-1 player assesses the outcome of *only* the considered move.





## Example 2: LA-1

- A lookahead-1 player assesses the outcome of *only* the considered move.

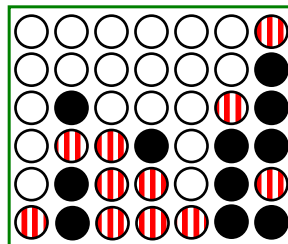
What scores change with the increased LA?  
***none of them!***

LA-1 scores for ●

50	50	50	50	50	50	-1
----	----	----	----	----	----	----

● 'X'  
● 'O'

●  
to  
move



## Example 2: LA-2

- A lookahead-2 player looks 2 moves ahead.
  - what if I make this move, and then my opponent makes *its best move*?
  - note:*** we assume the opponent looks ahead  $2 - 1 = 1$  move

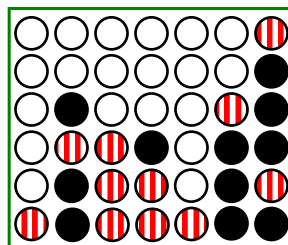
LA-1 scores for ●

50	50	50	50	50	50	-1
----	----	----	----	----	----	----

● 'X'  
● 'O'

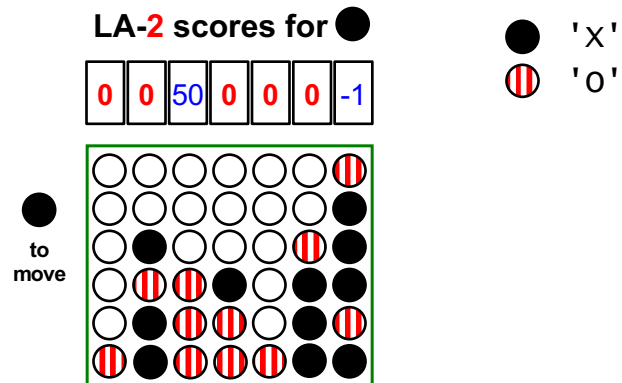
What would change?

●  
to  
move



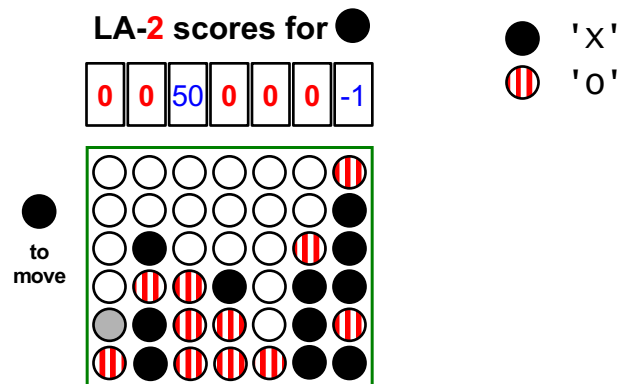
## Example 2: LA-2

- A lookahead-2 player looks 2 moves ahead.
  - what if I make this move, and then my opponent makes *its best move*?
  - **note:** we assume the opponent looks ahead  $2 - 1 = 1$  move



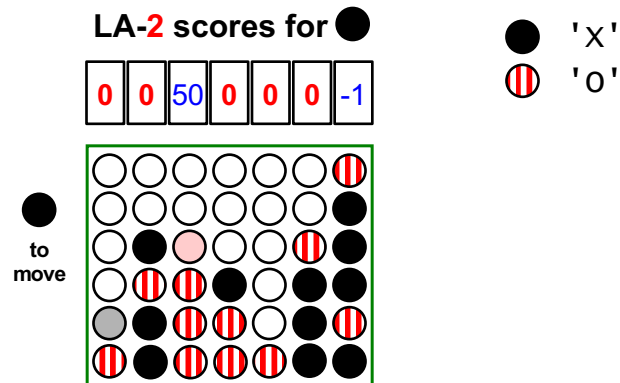
## Example 2: LA-2

- A lookahead-2 player looks 2 moves ahead.
  - what if I make this move, and then my opponent makes *its best move*?
  - **note:** we assume the opponent looks ahead  $2 - 1 = 1$  move



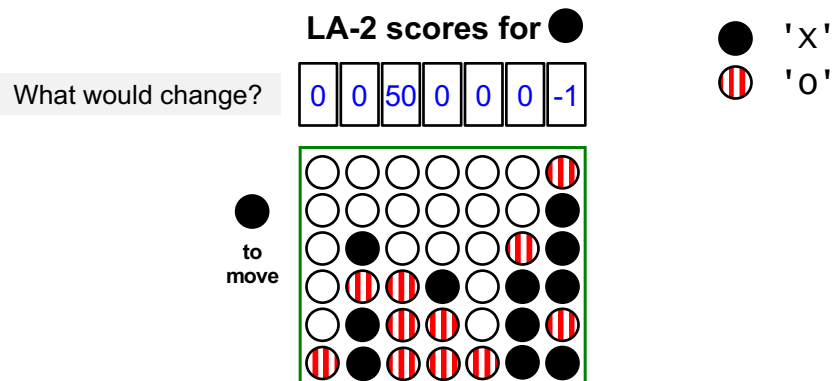
## Example 2: LA-2

- A lookahead-2 player looks 2 moves ahead.
  - what if I make this move, and then my opponent makes *its best move*?
  - note:** we assume the opponent looks ahead  $2 - 1 = 1$  move



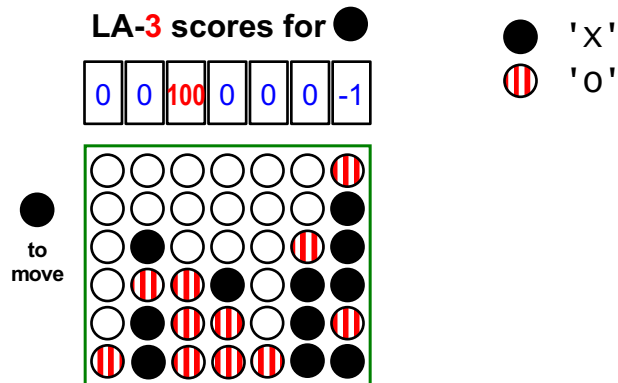
## LA-3!

- A lookahead-3 player looks 3 moves ahead.
  - what if I make this move, and then my opponent makes its best move, *and then I make my best subsequent move*?
  - note:** we assume the opponent looks ahead  $3 - 1 = 2$  moves



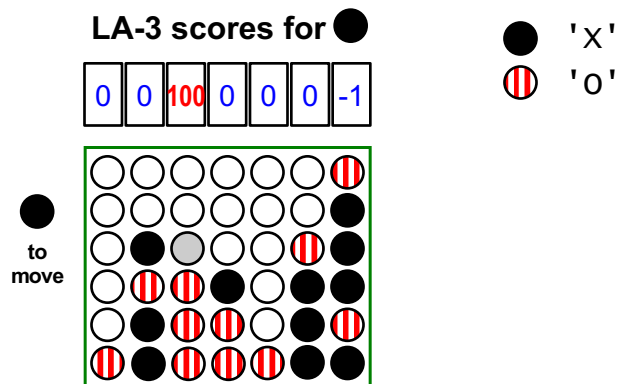
## LA-3!

- A lookahead-3 player looks 3 moves ahead.
  - what if I make this move, and then my opponent makes its best move, *and then I make my best subsequent move?*
  - **note:** we assume the opponent looks ahead  $3 - 1 = 2$  moves



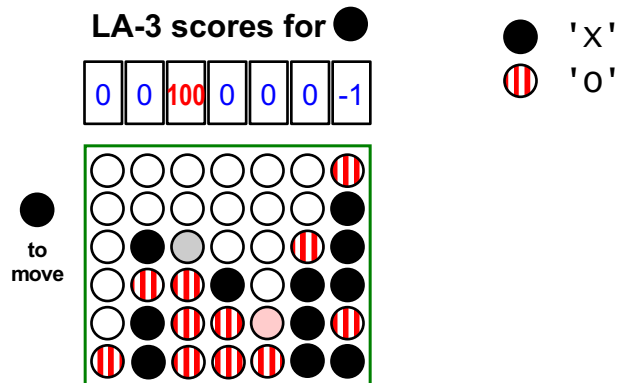
## LA-3!

- A lookahead-3 player looks 3 moves ahead.
  - what if I make this move, and then my opponent makes its best move, *and then I make my best subsequent move?*
  - **note:** we assume the opponent looks ahead  $3 - 1 = 2$  moves



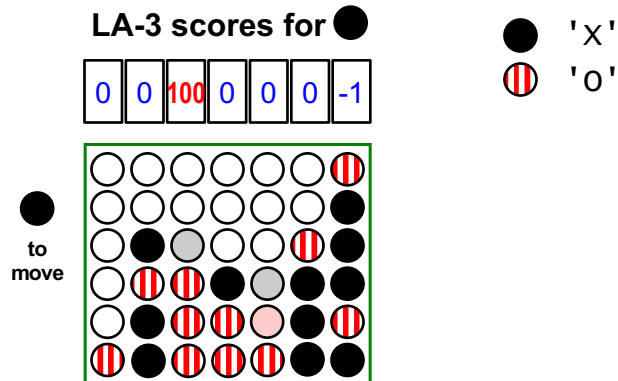
## LA-3!

- A lookahead-3 player looks 3 moves ahead.
  - what if I make this move, and then my opponent makes its best move, *and then I make my best subsequent move?*
  - **note:** we assume the opponent looks ahead  $3 - 1 = 2$  moves



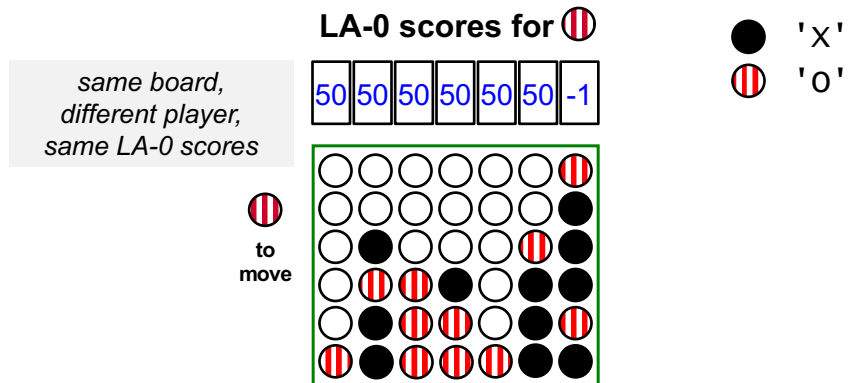
## LA-3!

- A lookahead-3 player looks 3 moves ahead.
  - what if I make this move, and then my opponent makes its best move, *and then I make my best subsequent move?*
  - **note:** we assume the opponent looks ahead  $3 - 1 = 2$  moves



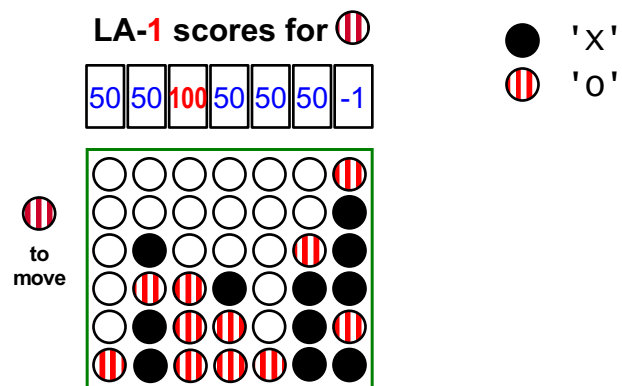
## Example 2: LA-0

- A lookahead-0 player only assesses the current board (0 moves!).



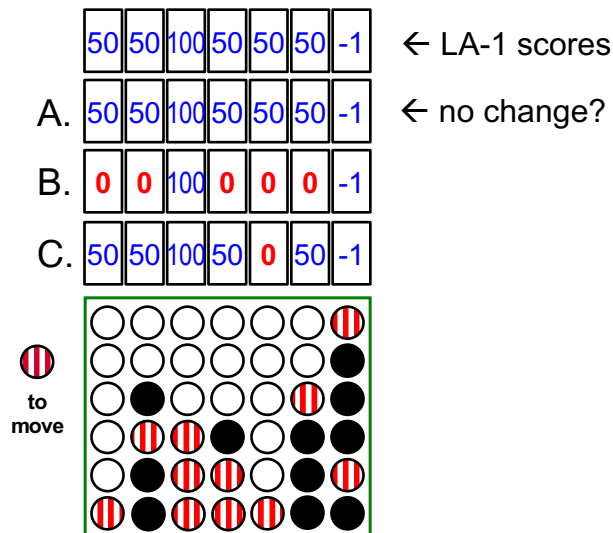
## Example 2: LA-1

- A lookahead-1 player assesses the outcome of *only* the considered move.



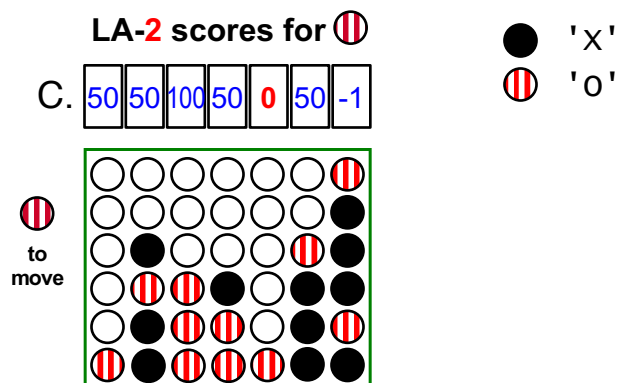
## What Are the LA-2 Scores for ?

- Look 2 moves ahead. Assume the opponent looks 1 move ahead.



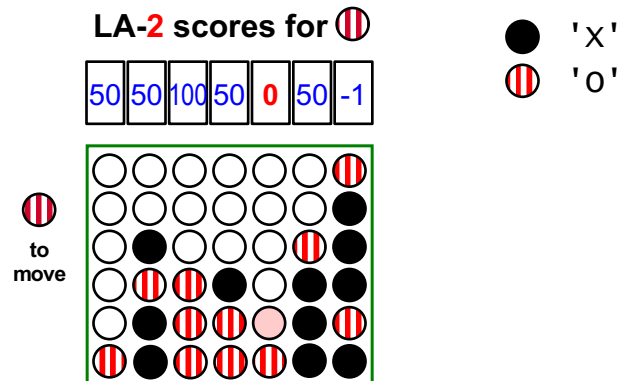
## Example 2: LA-2

- A lookahead-2 player looks 2 moves ahead.
  - what if I make this move, and then my opponent makes *its best move*?
  - note:** we assume the opponent looks ahead  $2 - 1 = 1$  move



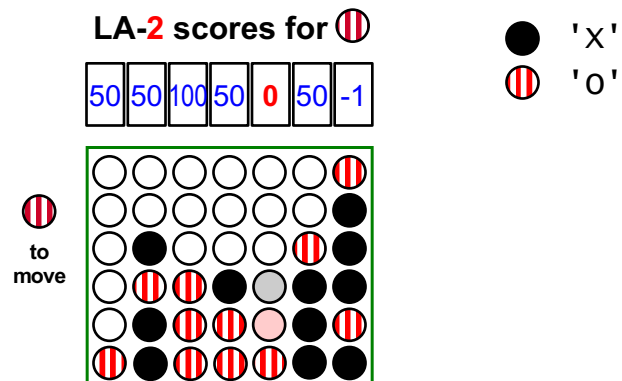
## Example 2: LA-2

- A lookahead-2 player looks 2 moves ahead.
  - what if I make this move, and then my opponent makes *its best move*?
  - **note:** we assume the opponent looks ahead  $2 - 1 = 1$  move



## Example 2: LA-2

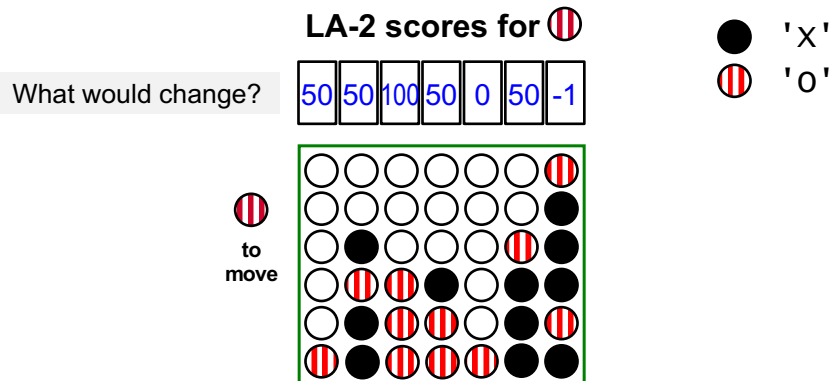
- A lookahead-2 player looks 2 moves ahead.
  - what if I make this move, and then my opponent makes *its best move*?
  - **note:** we assume the opponent looks ahead  $2 - 1 = 1$  move





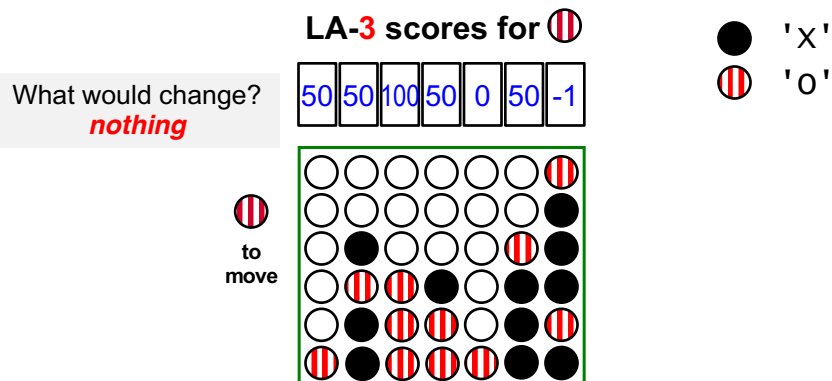
## Example 2: LA-3

- A lookahead-3 player looks 3 moves ahead.
  - what if I make this move, and then my opponent makes its best move, *and then I make my best subsequent move?*
  - **note:** we assume the opponent looks ahead  $3 - 1 = 2$  moves



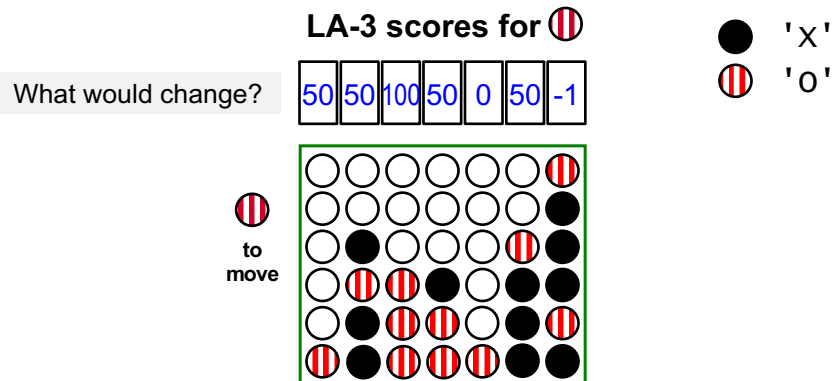
## Example 2: LA-3

- A lookahead-3 player looks 3 moves ahead.
  - what if I make this move, and then my opponent makes its best move, *and then I make my best subsequent move?*
  - **note:** we assume the opponent looks ahead  $3 - 1 = 2$  moves



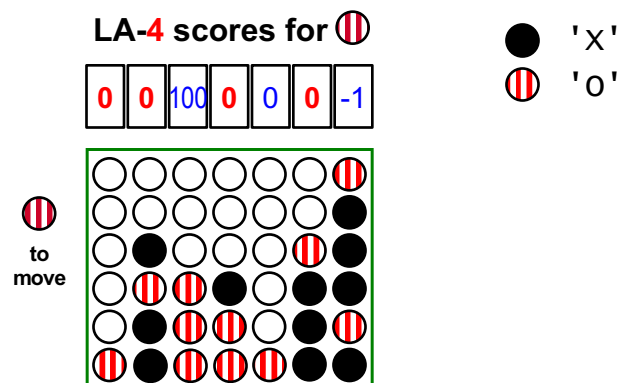
## LA-4!

- A lookahead-4 player looks 4 moves ahead.
  - assumes the opponent looks ahead  $4 - 1 = 3$  moves



## LA-4!

- A lookahead-4 player looks 4 moves ahead.
  - assumes the opponent looks ahead  $4 - 1 = 3$  moves





## LA-4!

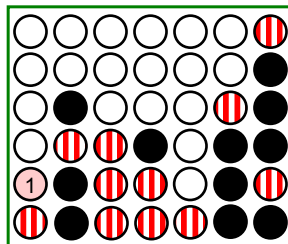
- A lookahead-4 player looks 4 moves ahead.
  - assumes the opponent looks ahead  $4 - 1 = 3$  moves

### LA-4 scores for

Consider column 0:  
1. 'O' moves there.

0	0	100	0	0	0	-1
---	---	-----	---	---	---	----

 'X'  
 'O'





## LA-4!

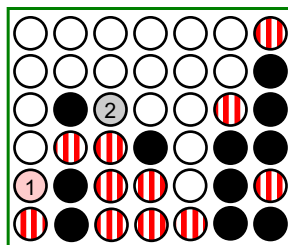
- A lookahead-4 player looks 4 moves ahead.
  - assumes the opponent looks ahead  $4 - 1 = 3$  moves

### LA-4 scores for

Consider column 0:  
1. 'O' moves there.  
2. 'X' moves to 2.

0	0	100	0	0	0	-1
---	---	-----	---	---	---	----

 'X'  
 'O'



## LA-4!

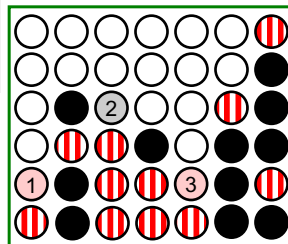
- A lookahead-4 player looks 4 moves ahead.
  - assumes the opponent looks ahead  $4 - 1 = 3$  moves


### LA-4 scores for

Consider column 0:

1. 'O' moves there.
2. 'X' moves to 2.
3. 'O' moves to 4 to block a diagonal win.

0	0	100	0	0	0	-1
---	---	-----	---	---	---	----



● 'X'  
 'O'

## LA-4!

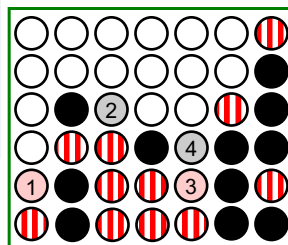
- A lookahead-4 player looks 4 moves ahead.
  - assumes the opponent looks ahead  $4 - 1 = 3$  moves


### LA-4 scores for

Consider column 0:

1. 'O' moves there.
2. 'X' moves to 2.
3. 'O' moves to 4 to block a diagonal win.
4. 'X' still wins horizontally!

0	0	100	0	0	0	-1
---	---	-----	---	---	---	----



● 'X'  
 'O'

## LA-4!

- A lookahead-4 player looks 4 moves ahead.
  - assumes the opponent looks ahead  $4 - 1 = 3$  moves

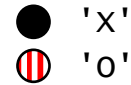
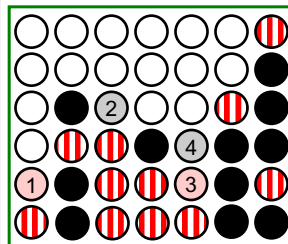
### LA-4 scores for 'O'

Consider column 0:

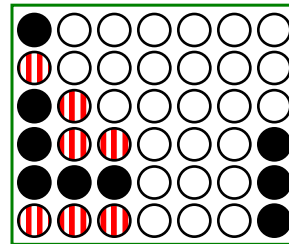
1. 'O' moves there.
2. 'X' moves to 2.
3. 'O' moves to 4 to block a diagonal win.
4. 'X' still wins horizontally!

Same thing holds for the other col's with new 0s.

0	0	100	0	0	0	-1
---	---	-----	---	---	---	----



## Solutions



'O'  
you - self - is  
playing 'o'



LA-0 scores for 'O':

Looks 0 moves into the future

col 0	col 1	col 2	col 3	col 4	col 5	col 6
-1	50	50	50	50	50	50

LA-1 scores for 'O':

Looks 1 move into the future

col 0	col 1	col 2	col 3	col 4	col 5	col 6

LA-2 scores for 'O':

Looks 2 moves into the future

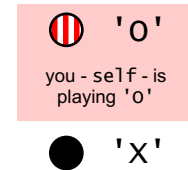
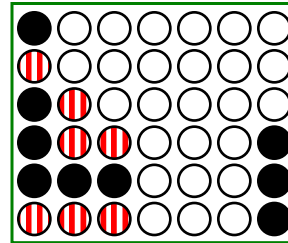
col 0	col 1	col 2	col 3	col 4	col 5	col 6

LA-3 scores for 'O':

Looks 3 moves into the future

col 0	col 1	col 2	col 3	col 4	col 5	col 6

## Solutions



	col 0	col 1	col 2	col 3	col 4	col 5	col 6
LA-0 scores for 'O': <small>Looks 0 moves into the future</small>	-1	50	50	50	50	50	50
LA-1 scores for 'O': <small>Looks 1 move into the future</small>	-1	50	50	100	50	50	50
LA-2 scores for 'O': <small>Looks 2 moves into the future</small>	-1	0	0	100	0	0	50
LA-3 scores for 'O': <small>Looks 3 moves into the future</small>	-1	0	0	100	0	0	100

## scores\_for – the AI in AIPlayer!

```
def scores_for(self, board):
    """ MUST return a list of scores – one for each column!!
    """
    scores = [50] * board.width
    for col in range(board.width):
```

???

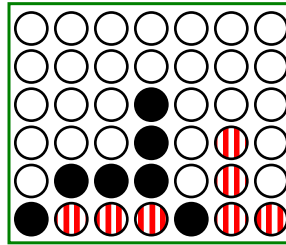
```
    return scores
```

Suppose you're playing  
with LA 2...

For each column:

- 1) add a checker to it
- 2) ask an opponent with  
LA 1 for its scores for the  
resulting board!
- 3) assume the opponent  
will makes its best move,  
and determine your  
score accordingly
- 4) remove checker!

scores\_for



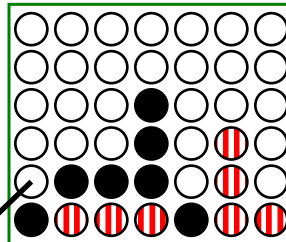
(self) 'X' ●  
possible  
next move ●

Suppose you're playing  
with LA 2...

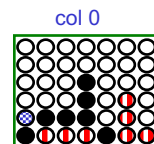
For each column:

- 1) add a checker to it
- 2) ask an opponent with  
LA 1 for its scores for the  
resulting board!
- 3) assume the opponent  
will makes its best move,  
and determine your  
score accordingly
- 4) remove checker!

scores\_for



(self) 'X' ●  
possible  
next move ●

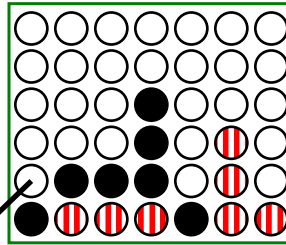


Suppose you're playing  
with LA 2...

For each column:

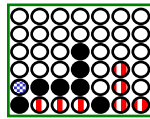
- 1) add a checker to it
- 2) ask an opponent with  
LA 1 for its scores for the  
resulting board!
- 3) assume the opponent  
will makes its best move,  
and determine your  
score accordingly
- 4) remove checker!

scores\_for



(self) 'X' ●  
possible  
next move ●

col 0



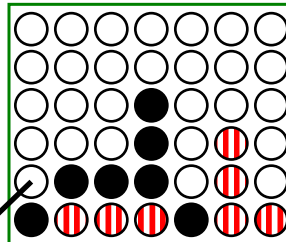
opp\_scores = [0,0,0,0,0,0,0]

Suppose you're playing  
with LA 2...

For each column:

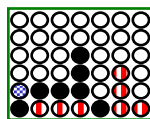
- 1) add a checker to it
- 2) ask an opponent with  
LA 1 for its scores for the  
resulting board!
- 3) assume the opponent  
will makes its best move,  
and determine your  
score accordingly
- 4) remove checker!

scores\_for



(self) 'X' ●  
possible  
next move ●

col 0



opp\_scores = [0,0,0,0,0,0,0]

max(opp\_scores) = 0

scores[0] = ?



Suppose you're playing  
with LA 2...

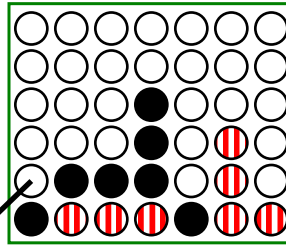
For each column:

- 1) add a checker to it
- 2) ask an opponent with  
LA 1 for its scores for the  
resulting board!

3) assume the opponent  
will make its best move,  
and determine your  
score accordingly

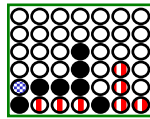
- 4) remove checker!

scores\_for



(self) 'X' ●  
possible  
next move ●

col 0



opp\_scores = [0,0,0,0,0,0,0]  
max(opp\_scores) = 0  
scores[0] = 100  
*A loss for my opponent  
is a win for me!*

Suppose you're playing  
with LA 2...

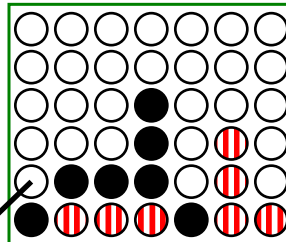
For each column:

- 1) add a checker to it
- 2) ask an opponent with  
LA 1 for its scores for the  
resulting board!

3) assume the opponent  
will make its best move,  
and determine your  
score accordingly

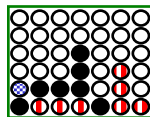
- 4) remove checker!

scores\_for

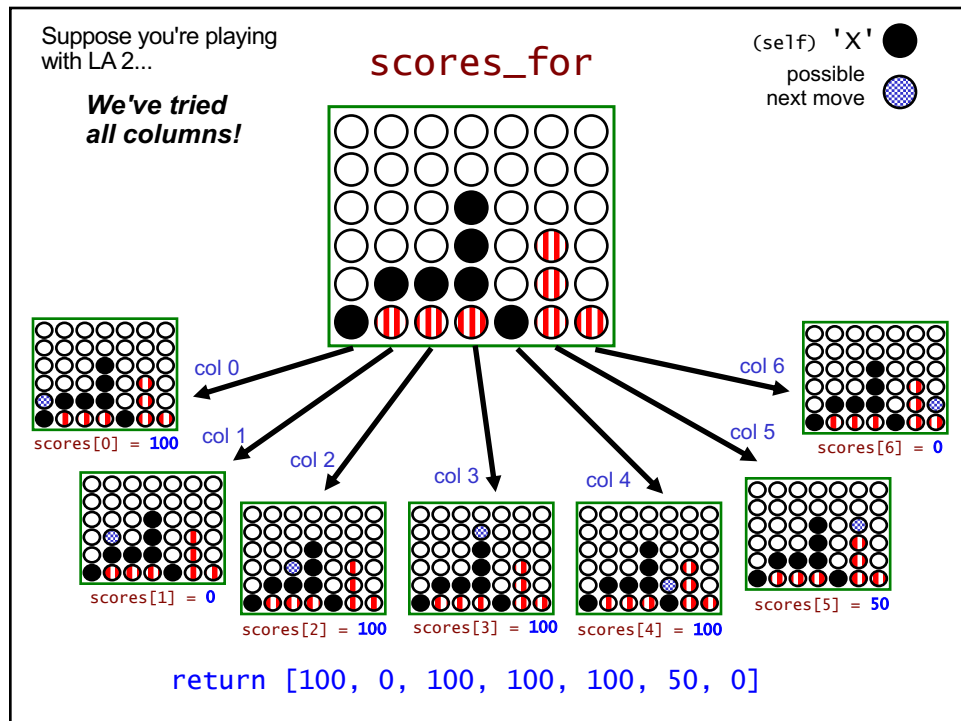


(self) 'X' ●  
possible  
next move ●

col 0



opp\_scores = [0,0,0,0,0,0,0]  
max(opp\_scores) = 0  
scores[0] = 100  
*A loss for my opponent  
is a win for me!*



## scores\_for – the AI in AIPlayer!

```
def scores_for(self, board):
    """ MUST return a list of scores – one for each column!!
    """
    scores = [50] * board.width
    for col in range(board.width):

        ???

    return scores
```

## scores\_for – the AI in AIPlayer!

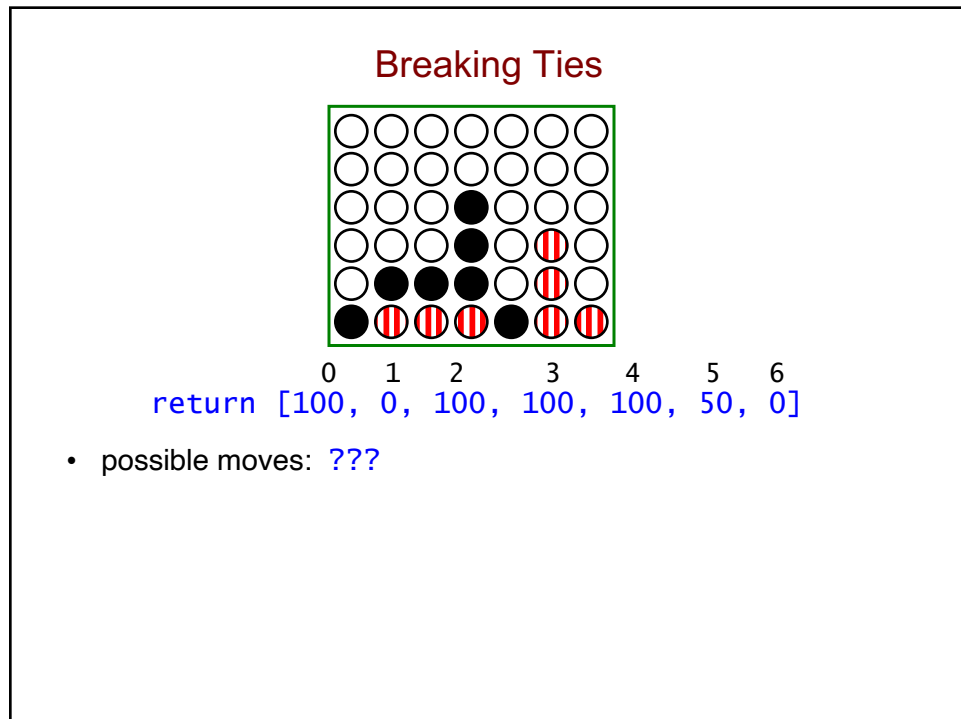
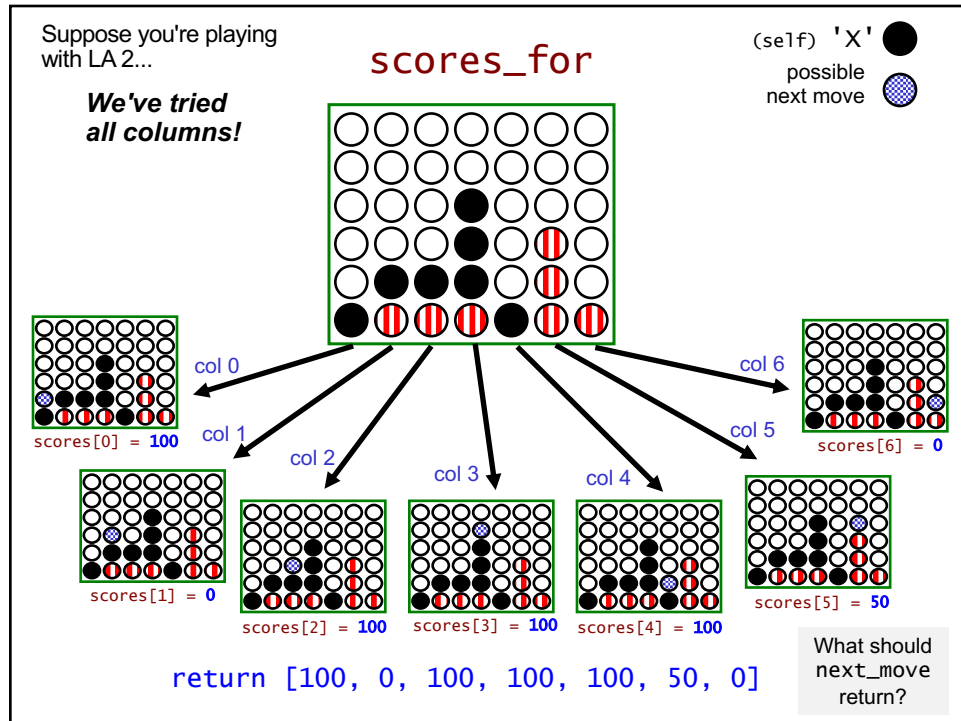
```
def scores_for(self, board):
    """ MUST return a list of scores – one for each column!!
    """
    scores = [50] * board.width
    for col in range(board.width):
        if col is full:
            use -1 for scores[col]
        elif already win/loss:
            use appropriate score (100 or 0)
        elif lookahead is 0:
            use 50
        else:
            try col – adding a checker to it
            create an opponent with self.lookahead – 1
            opp_scores = opponent.scores_for(...)
            scores[col] = ???
            remove checker

    return scores
```

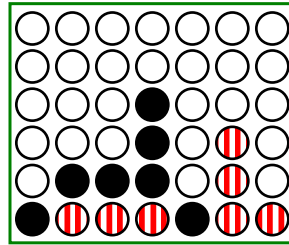
## scores\_for – the AI in AIPlayer!

```
def scores_for(self, board):
    """ MUST return a list of scores – one for each column!!
    """
    scores = [50] * board.width
    for col in range(board.width):
        if col is full:
            use -1 for scores[col]
        elif already win/loss:
            use appropriate score (100 or 0)
        elif lookahead is 0:
            use 50
        else:
            try col – adding a checker to it
            create an opponent with self.lookahead – 1
            opp_scores = opponent.scores_for(...)
            scores[col] = ???
            remove checker

    return scores
```



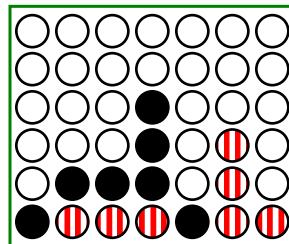
### Breaking Ties



return [100, 0, 100, 100, 100, 50, 0]

- possible moves: [0, 2, 3, 4]

### Breaking Ties



return [100, 0, 100, 100, 100, 50, 0]

- possible moves: [0, 2, 3, 4]
- self.tiebreak == 'LEFT': return 0
- self.tiebreak == 'RIGHT': return 4
- self.tiebreak == 'RANDOM': choose at random!