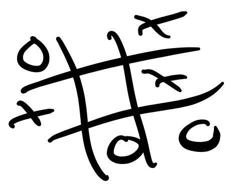
orig Board = [0, 1, ...]



1. mm (origBoard, ai Player)

- List of empty = [1, 4, 6]

  - Check for terminal states, & loop

ai p - newboard = [0, X, X, X, 4, X, 6, 0, 0]

- Calls itself with newBoard, huplayer

mm (newBoard, huplayer)

.... wait for value from this 2nd FC

2. - List of empty [4, 6]

  - Check for terminal states & loop

huplayer - newboard = [0, X, X, X, O, X, 6, 0, 0]

- calls itself w/ mm(newBoard, aiPlayer)

... Wait for FC value

3. List [6]

  - Check terminal

The algorithm now finds a win for the human player, so it returns a property of -10

4. - List of empty [X, 6]

- change newboard to

newboard = [0, X, X, X, 4, X, O, 0, 0]

- call mm(newBoard, aiplayer)

... wait for func value

5. List of empty [4]

- Finds win for human after checking for terminal states

- returns −10

6. As a result, First FC has evaluated first spot as having value of −10. Next it changes [newBoard] by placing aiPlayer in the 2nd spot. Then

it calls itself w/ newBoard &
human player.

- Score of [1,4,6]

$\swarrow$

= -10.

- now evaluate 4.

---

7. List [1, 4, 6]

- Check for terminal states, & loop
- newboard = [0, 1, x, x, x, x, 6, 0, 0]

- Found win for ai player,
return $\underline{+10}$

---

8. List [1, 4, 6]

- Check for terminal states, & loop

- newboard = [0, 1, x, x, 4, x,

- no terminal

- mm(newBoard, humanp)

---

9. List [1, 4]

- Check for terminal states, & loop
- newboard = [0, 0, x, x, 4, X, X, 0, 0]

- mm(newBoard, cip)

---

10. List [4]

- Check for terminal states, & loop
- newboard = [0, 0, x, x, x, X, X, 0, 0]

mm(nB, hp)

---

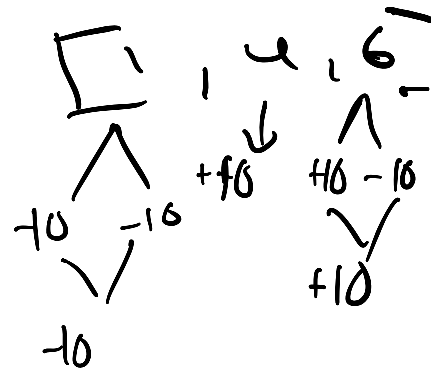
11. List [ ]

- Check for terminal -

Terminal Found

Win for ai

+10


[1 , 4 , 6]
-10 -10
-10
+10
+10 -10
+10