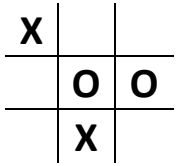
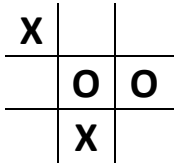
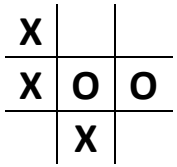
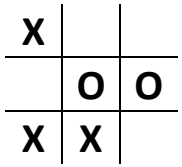
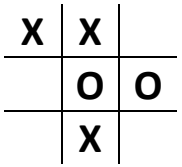
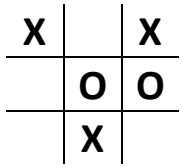

X
O O
X

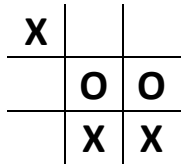

X
O O
X


X
X O O
X


X
O O
X X


X X
O O
X


X X
O O
X


X
O O
X X

X		
	O	O
	X	

X		
X	O	O
	X	

X		
	O	O
X	X	

X	X	
	O	O
	X	

X		X
	O	O
	X	

X		
	O	O
	X	X

X		
O	O	O
X	X	

L

-1

X	X	
O	O	O
	X	

L

-1

X		X
O	O	O
	X	

L

-1

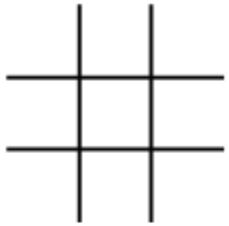
X		
O	O	O
	X	X

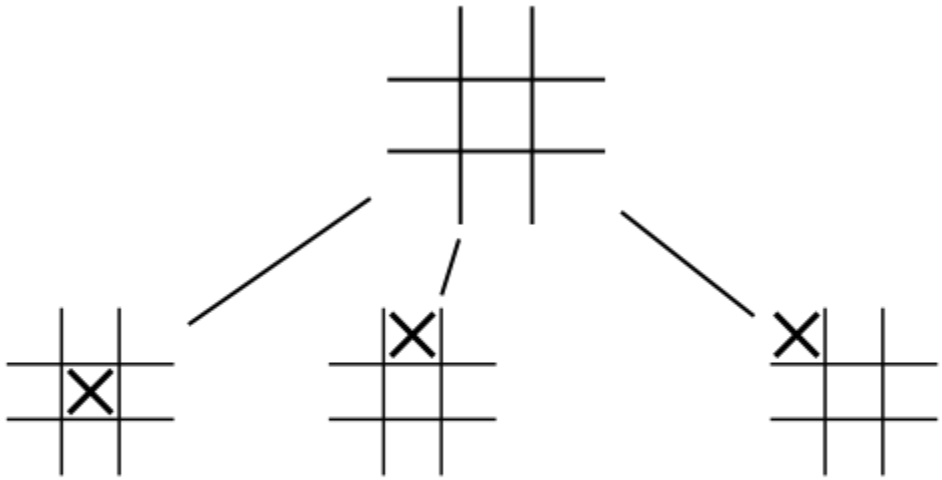
L

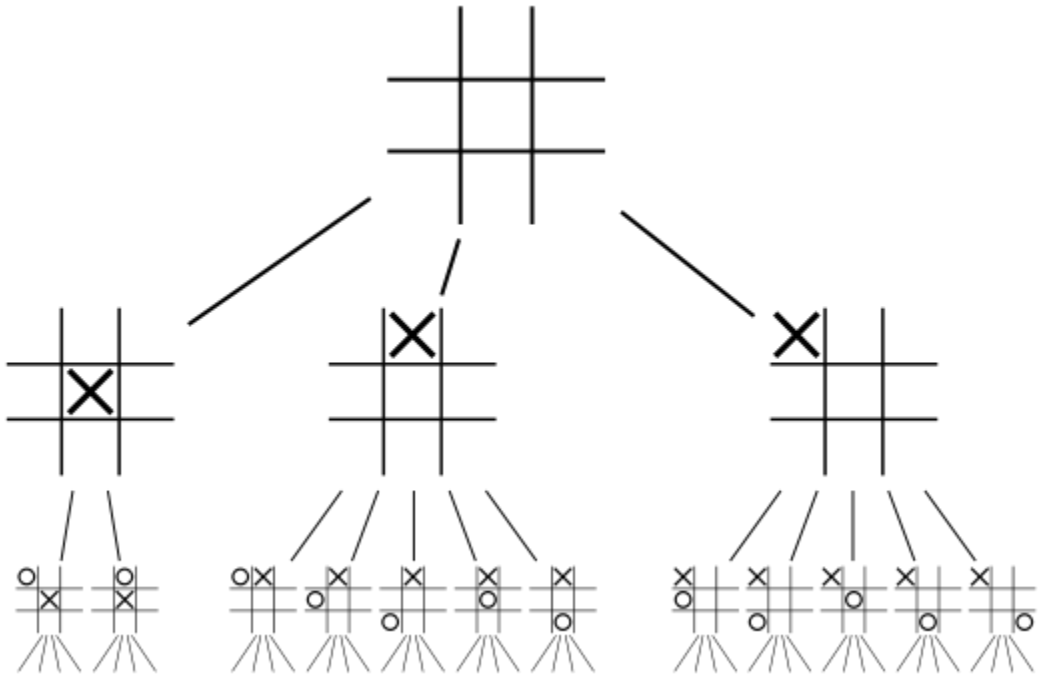
-1

Adversarial Search / Game Tree Search / Minimax Search

```
minimax(state,game,player):
    if game.win(state,"me") then return +1
    else if game.win(state,"opponent") then return -1
    else if game.tie(state) then return 0
    else if player="me"
        then max {minimax(s',game,"opponent") | a ∈ game.actions(state) and s' = child(a,state)}
    else min {minimax(s',game,"me") | a ∈ game.actions(state) and s' = child(a,state)}
```





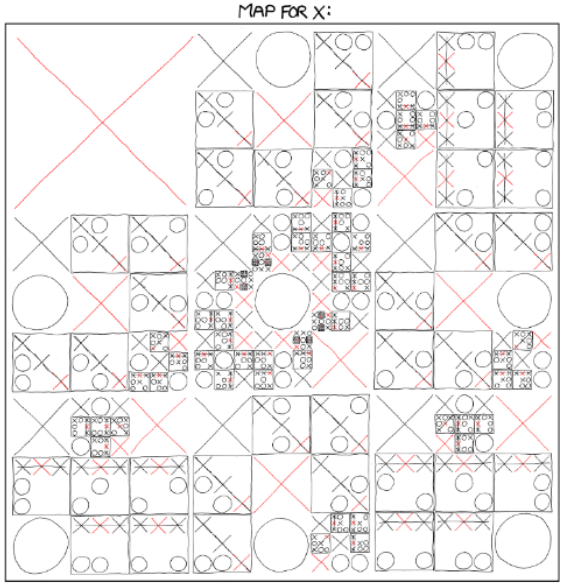


(xkcd on tic tac toe)

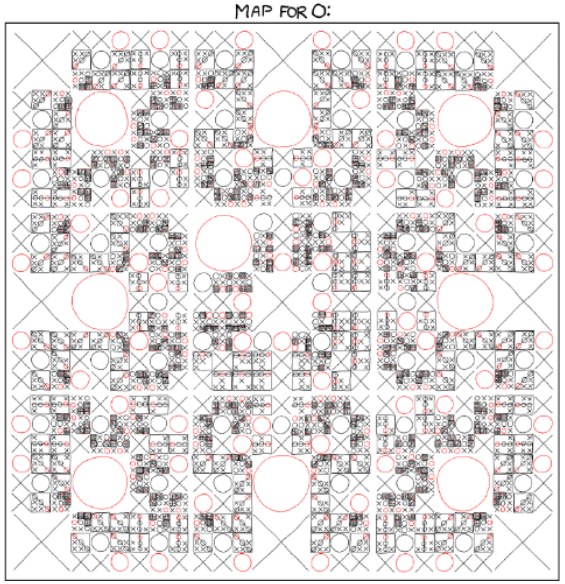
COMPLETE MAP OF OPTIMAL TIC-TAC-TOE MOVES

YOUR MOVE IS GIVEN BY THE POSITION OF THE LARGEST RED SYMBOL ON THE GRID. WHEN YOUR OPPONENT PICKS A MOVE, ZOOM IN ON THE REGION OF THE GRID WHERE THEY WENT. REPEAT.

MAP FOR X:



MAP FOR O:



$$f(P) = 200(K-K') + 9(Q-Q') + 5(R-R') + 3(B-B'+N-N') + (P-P') - 0.5(D-D'+S-S'+I-I') + 0.1(M-M') + \ldots$$

Informed Adversarial Search / Game Tree Search / Minimax Search

```
minimax(state,game,player,depth):
    if game.win(state,”me”) then return +1
    else if game.win(state,”opponent”) then return -1
    else if game.tie(state) then return 0
    else if depth=0 then return game.f(state)
    else if player=”me”
        then max {minimax(s’,game,”opponent”,depth-1) | a ∈ game.actions(state) and s’ = child(a,state)}
    else min {minimax(s’,game,”me”,depth-1) | a ∈ game.actions(state) and s’ = child(a,state)}
```