main

Diapositivas-Ing.-Espino-IA1 / MD / Lección 5. Búsquedas por adversario.md

200 lines (200 loc) · 14 KB

5. BÚSQUEDAS POR ADVERSARIO

M.SC. LUIS FERNANDO ESPINO BARRIOS

2020

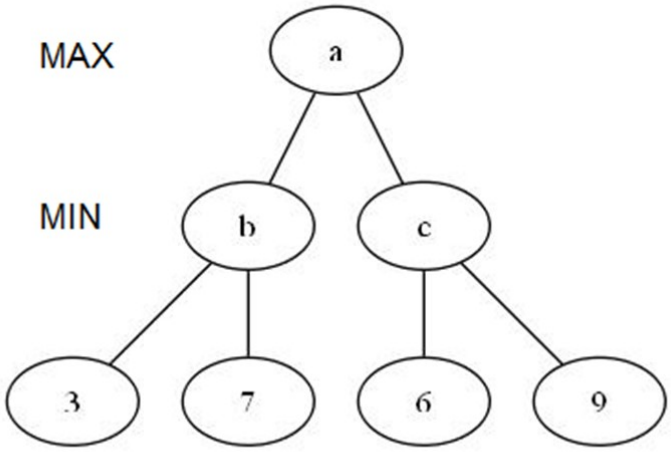
BÚSQUEDAS POR ADVERSARIO

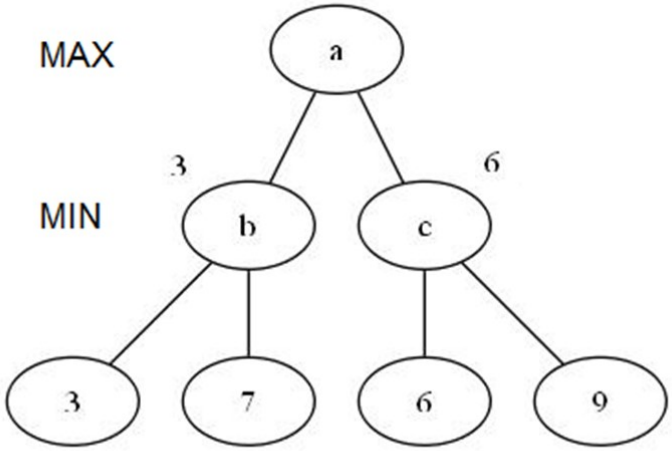
- Plantea un problema donde intervienen dos o más jugadores, sean personas o máquinas, y el objetivo es adelantarse a posibles movimientos para seleccionar adecuadamente el movimiento propio.

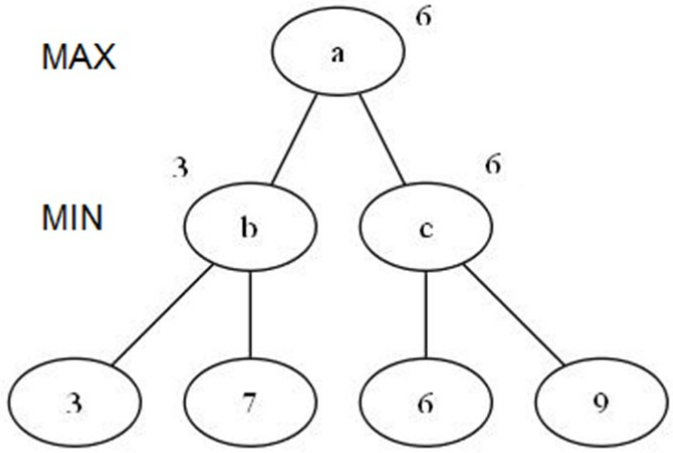
MINIMAX

- La idea principal del algoritmo es iniciar en una posición y utilizar un generador de movimientos plausibles para generar un conjunto de posiciones de posibles sucesores, se aplica una función de evaluación estática y se selecciona el mejor movimiento promoviendo el valor.

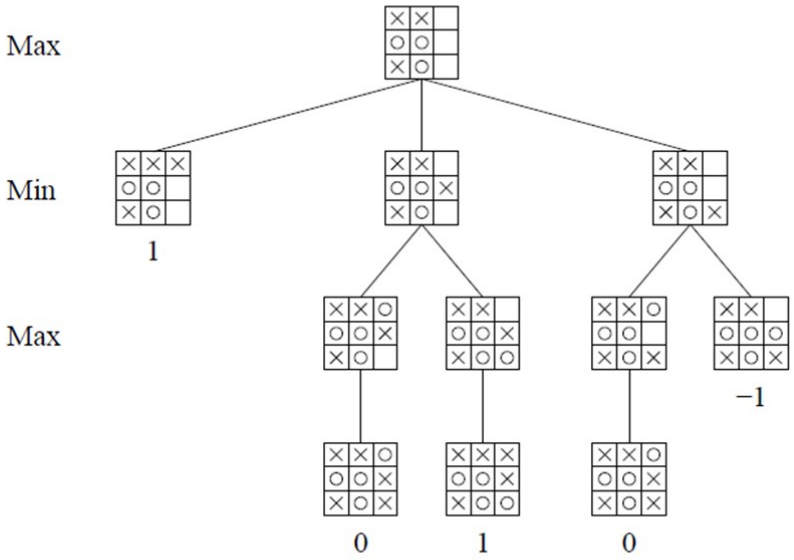
EJEMPLO

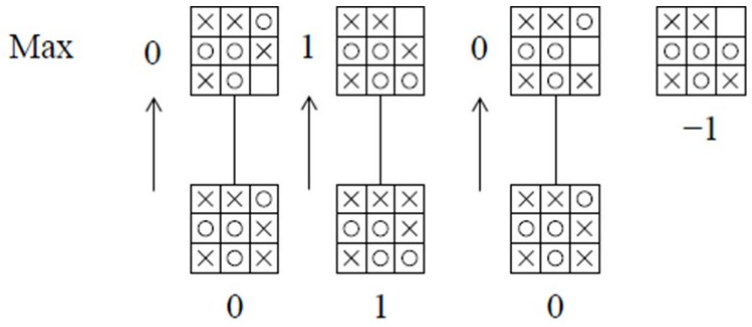


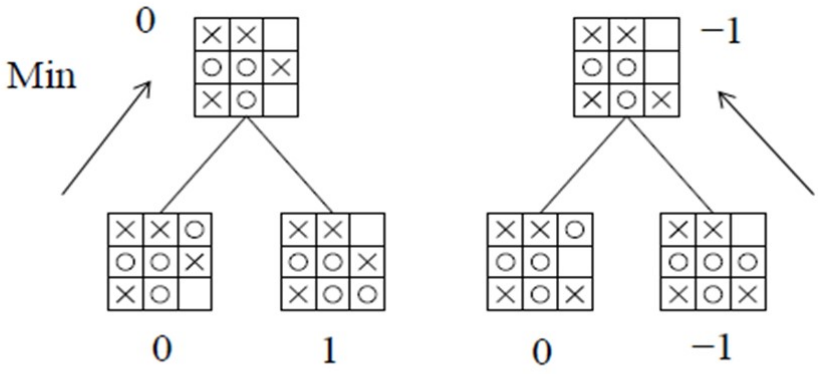

MAX
a
3
6
MIN
b
c
3
7
6
9

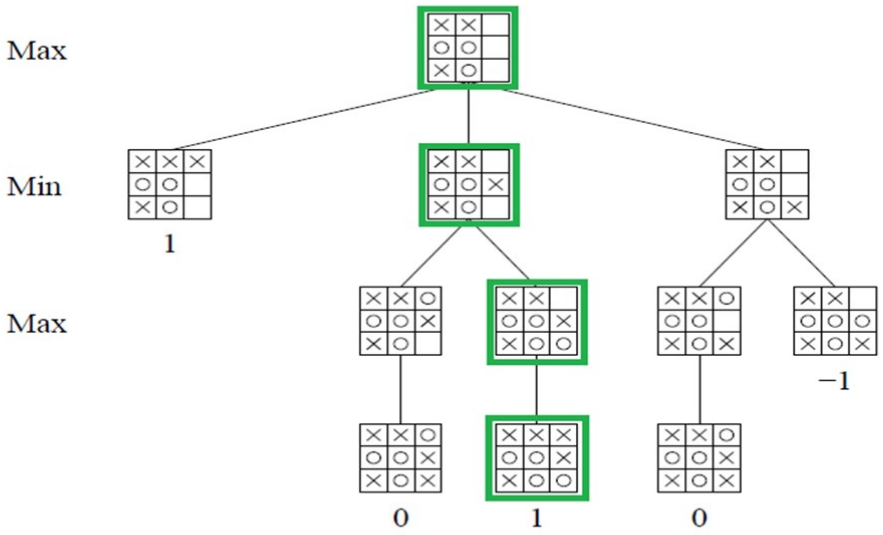

6
MAX
a
3
6
MIN
b
c
3
7
6
9

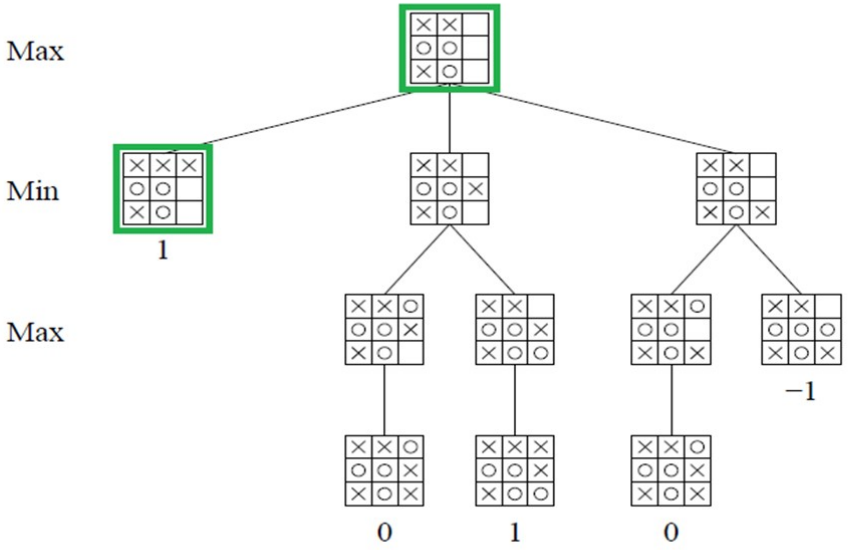
EJEMPLO DE TRES EN RAYA



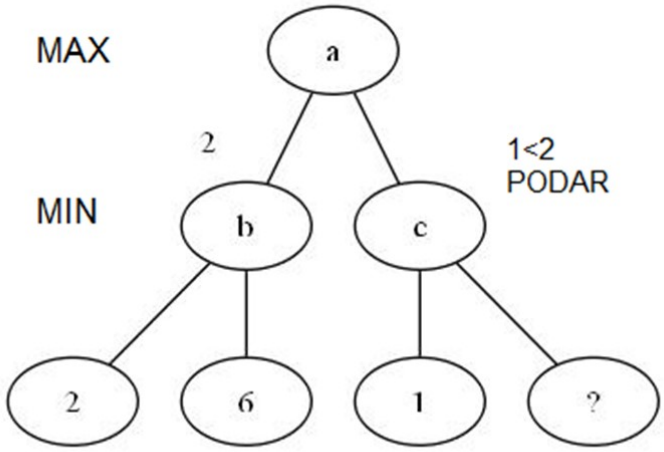

Max
0
1
0
−1
0
1
0


0
Min
−1
0
1
0
−1


Max
Min
Max
1
0
1
0
−1


Max
Min
Max
1
0
1
0
−1

PODA ALFA BETA

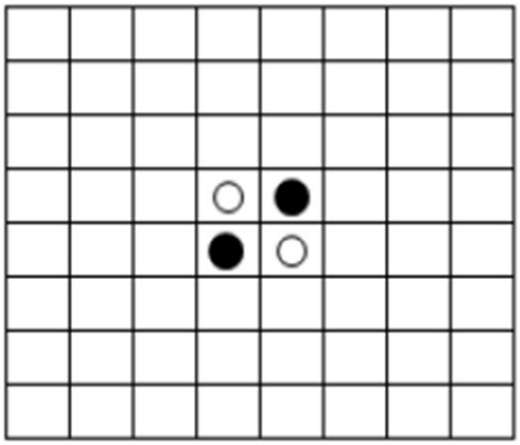


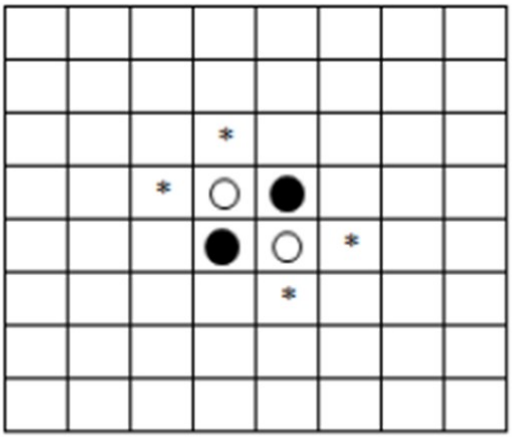
EJERCICIO

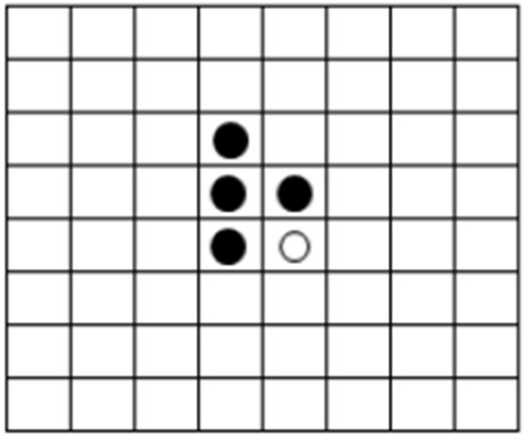
¿Qué heurísticas se deben utilizar para los juegos damas, ajedrez, 2048?

EJERCICIO

- Analizar el juego Reversi.







```
'(( 120 -20 20  5  5 20 -20 120 )
  ( -20 -40 -5 -5 -5 -5 -40 -20 )
  (  20  -5 15  3  3 15  -5  20 )
  (   5  -5  3  3  3  3  -5   5 )
  (   5  -5  3  3  3  3  -5   5 )
  (  20  -5 15  3  3 15  -5  20 )
  ( -20 -40 -5 -5 -5 -5 -40 -20 )
  ( 120 -20 20  5  5 20 -20 120 ))))
```

EJERCICIO

- Explicar qué hace el siguiente código en Python

```
import random
import string

def succ(node):
    return [[random.choice(string.ascii_uppercase),random.randint(0,20),""],
            [random.choice(string.ascii_uppercase),random.randint(0,20),""],
            [random.choice(string.ascii_uppercase),random.randint(0,20),""]]

def minimax(node, depth, m):
    if depth == 0:
        return node
    if m:

        temp = succ(node[0])
        print(temp)
        maxi = -999
        while temp:
            child = temp.pop(0)
            h = minimax(child, depth-1, False)
            if h[1]>maxi:
                best = child
                maxi = best[1]
        node[1]=best[1]
        node[2]=best[0]
        return node
    else:
        temp = succ(node[0])
        mini = 999
        print(temp)
        while temp:
            child = temp.pop(0)
            h = minimax(child, depth-1, True)
            if h[1]<mini:
                best = child
                mini = best[1]
        node[1]=best[1]
        node[2]=best[0]
        return node

print(minimax (["A",0,""], 2, True))
```

TAREA 5

- Hacer grupos de 3 estudiantes
- A partir de 3 proveedores de servicios en la nube investigar características para tener una instancia donde se ejecute Python con un nombre de dominio y compararlos. (Free Tier)
- Seleccionar un proveedor, investigar y describir con qué framework trabaja Python en dicho proveedor para el ambiente web.

FIN