ALEJANDRO SANTORUM VARELA   EJERCICIOS HOJA 1   44090946-S

1.

a) HEURÍSTICA MONÓTONA PROPUESTA: número de fallos con la secuencia del estado final. Se considerará un fallo que la letra no esté en el lugar de la secuencia correcto o que directamente no esté presente en la secuencia.

Ejemplos: $h('RAM') = 0$ $h('MAR') = 2$ $h('RAV') = 1$

$h('AMR') = 3$ $h('MVR') = 3$ $h('RVM') = 1$

Demostración de monotonía: Intentemos encontrar un contraejemplo tal que $h(n) > \Gamma_{n \to n'} + h(n')$ siendo n' sucesor de n.

Supongamos que $h(n) = 3$ (peor caso) entonces hay 3 posibles acciones para derivar en un estado sucesor: (3 colores mal colocados)

- (1) Coste = 1, e.d., $\Gamma_{n \to n'} = 1$

  En esta acción se dejan dos colores fijos, por lo que, como máximo, sólo vamos a poder poner un color bien.

  $\Rightarrow \Gamma_{n \to n'} = 1 \wedge h(n') = 2$ (en el mejor caso)

  $\Rightarrow h(n) = 3 \wedge \Gamma_{n \to n'} + h(n') = 3 \Rightarrow$ no hay contraejemplo escogiendo acción (1)

- (2) Coste = 2, e.d., $\Gamma_{n \to n'} = 2$

  En esta acción se deja un color fijo, por lo que, como máximo, sólo vamos a poder colocar bien dos colores.

  $\Rightarrow \Gamma_{n \to n'} = 2 \wedge h(n') = 1$ (en el mejor caso)

  $\Rightarrow h(n) = 3 \wedge \Gamma_{n \to n'} + h(n') = 3 \Rightarrow$ no hay contraejemplo escogiendo acción (2)

- (3) Coste = 3, e.d., $\Gamma_{n \to n'} = 3$

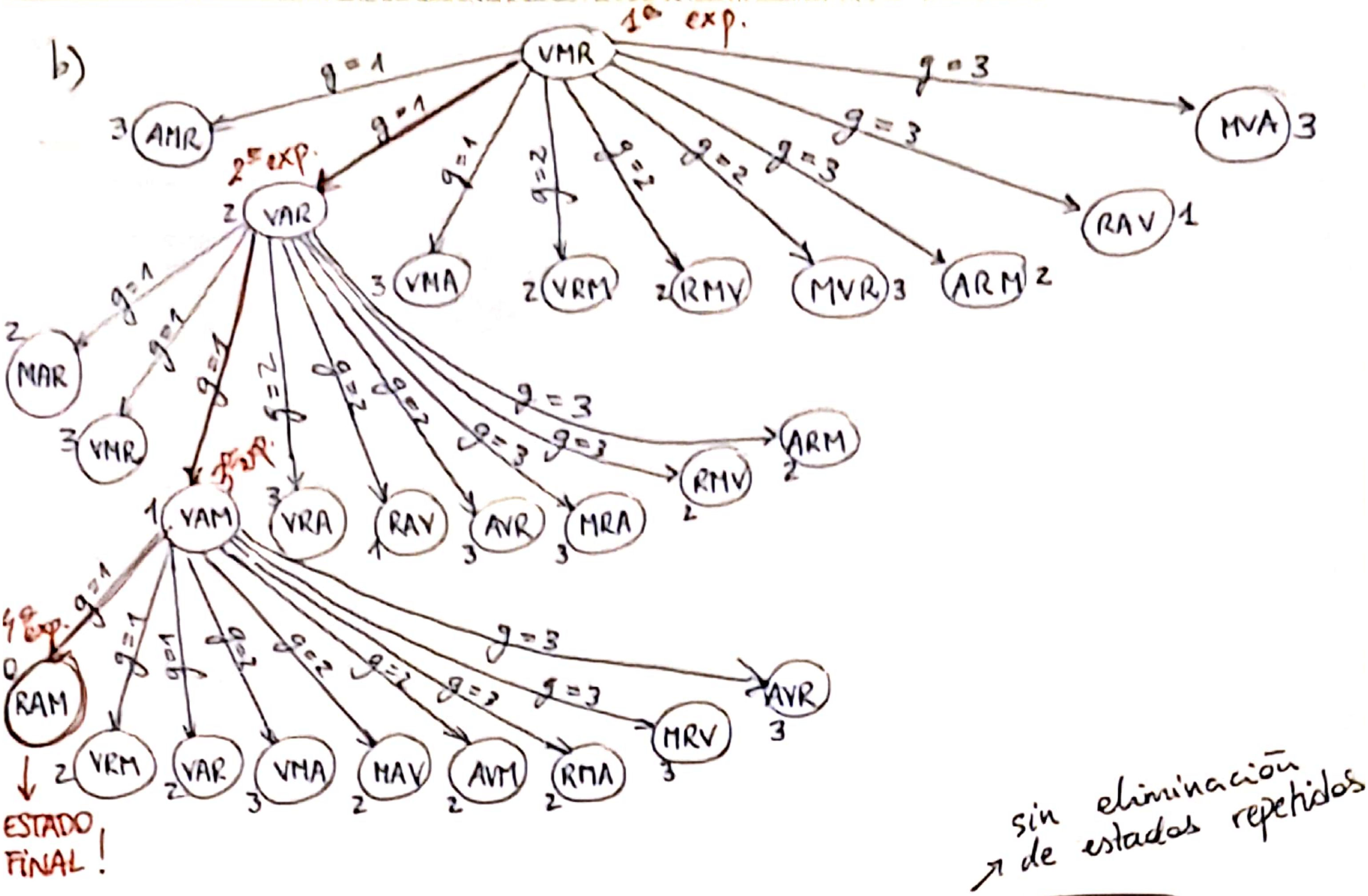
  En esta acción no se dejan colores fijos por lo que se podría dar la situación de llegar a la secuencia final.

  $\Rightarrow \Gamma_{n \to n'} = 3 \wedge h(n') = 0$ (en el mejor caso)

  $\Rightarrow h(n) = 3 \wedge \Gamma_{n \to n'} + h(n') = 3 \Rightarrow$ no hay contraejemplo escogiendo acción (3)

$\Rightarrow \nexists$ contraejemplo $\Rightarrow$ la heurística propuesta es monótona.

b)



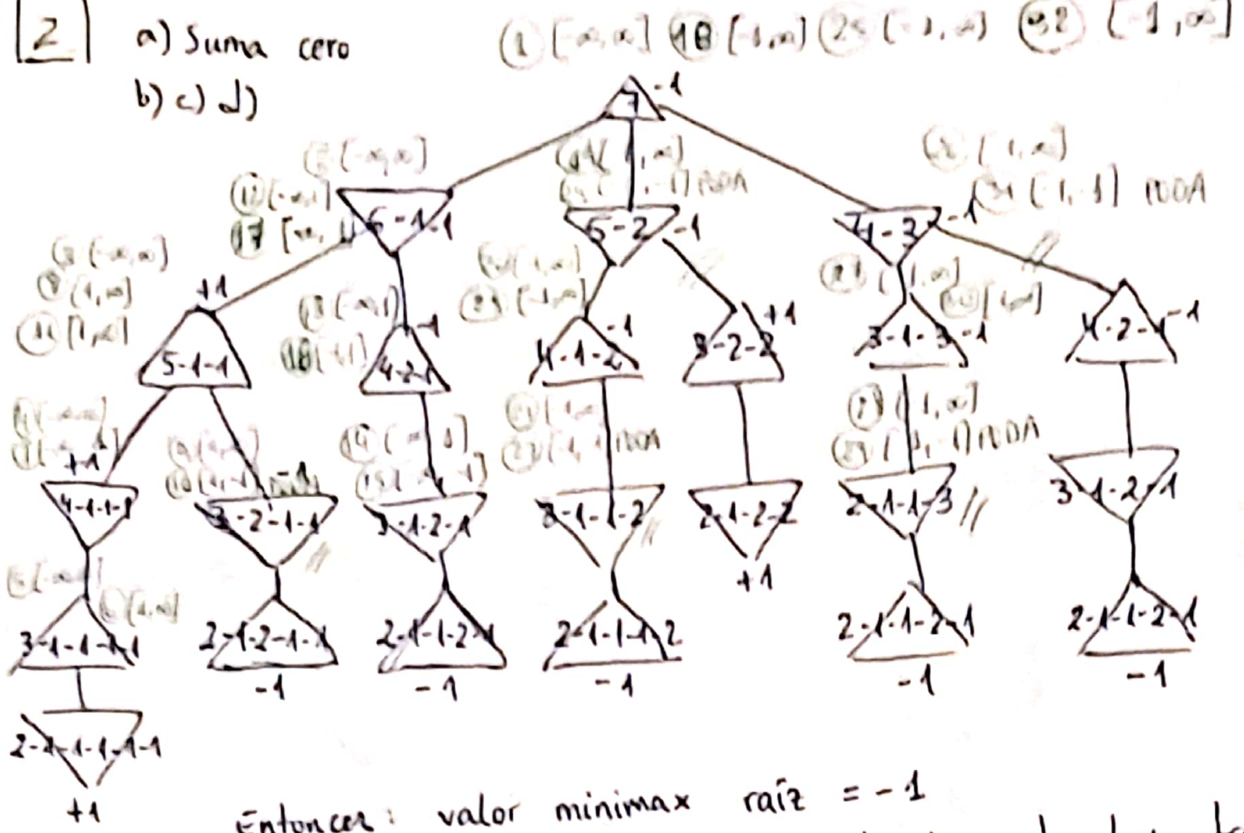
sin eliminación de estados repetidos

c) Tal y como hemos visto en clase, búsqueda en árbol + heurística admisible $\Rightarrow$ A^* óptima y completa. Como la heurística propuesta es monótona y monotonía $\Rightarrow$ admisibilidad $\Rightarrow$ A^* sin eliminación de estados repetidos con la heurística propuesta <u>es</u> óptima.

d) Tal y como hemos visto en clase, búsqueda en grafo + heurística monótona $\Rightarrow$ A^* óptima y completa. Como la heurística propuesta es monótona $\Rightarrow$ A^* con elim. de estados repetidos con la heurística propuesta <u>es</u> óptima.

2 a) Suma cero

b) c) d)



Entonces: valor minimax raíz $= -1$

Jugada óptima para MAX: cualquiera de las tres proporciona el mismo resultado.

3. a) Como el equipo de fumigación solo conoce la estructura del bosque (nada de la plaga) los estados de búsqueda serían:

NO(L) := noroeste limpio	NO(Pb) := " con plaga baja	NO(Pa) := " con plaga alta	NO(Pa+) := " " " " " alta
NE(L) := noreste limpio	NE(Pb) := " " " "	NE(Pa) := " " " "	NE(Pa+) := " " " " "
SE(L) := sureste limpio	SE(Pb) := " " " "	SE(Pa) := " " " "	SE(Pa+) := " " " "
SO(L) := suroeste limpio	SO(Pb) := " " " "	$\underbrace{\text{SO(Pa) := " " " "}}_{\text{falta 1 limpieza}}$	$\underbrace{\text{SO(Pa+) := " " "}}_{\text{2 limpiezas}}$

b) Acción 1: traslado → costes → horizontal: 2h; vertical: 2h; diagonal: 3h

→ reglas → traslados a estados no actuales; no es posible trasladarse a pie al SE

Acción 2: limpiar

↳ costes: cada limpieza 1h. Separación de 4h con plaga alta.

c) ESTADO INICIAL: Idealmente cualquier sector del bosque. Como SE no es accesible por tierra, lo ponemos como estado inicial para ser accedido por aire.

d) TEST OBJETIVO: haber pasado por todos los sectores (comprobando su estado) y haberlos limpiado por completo los que estuvieran infectados.

e) $h(n) \equiv$ nº de estados del bosque aún sin visitar:

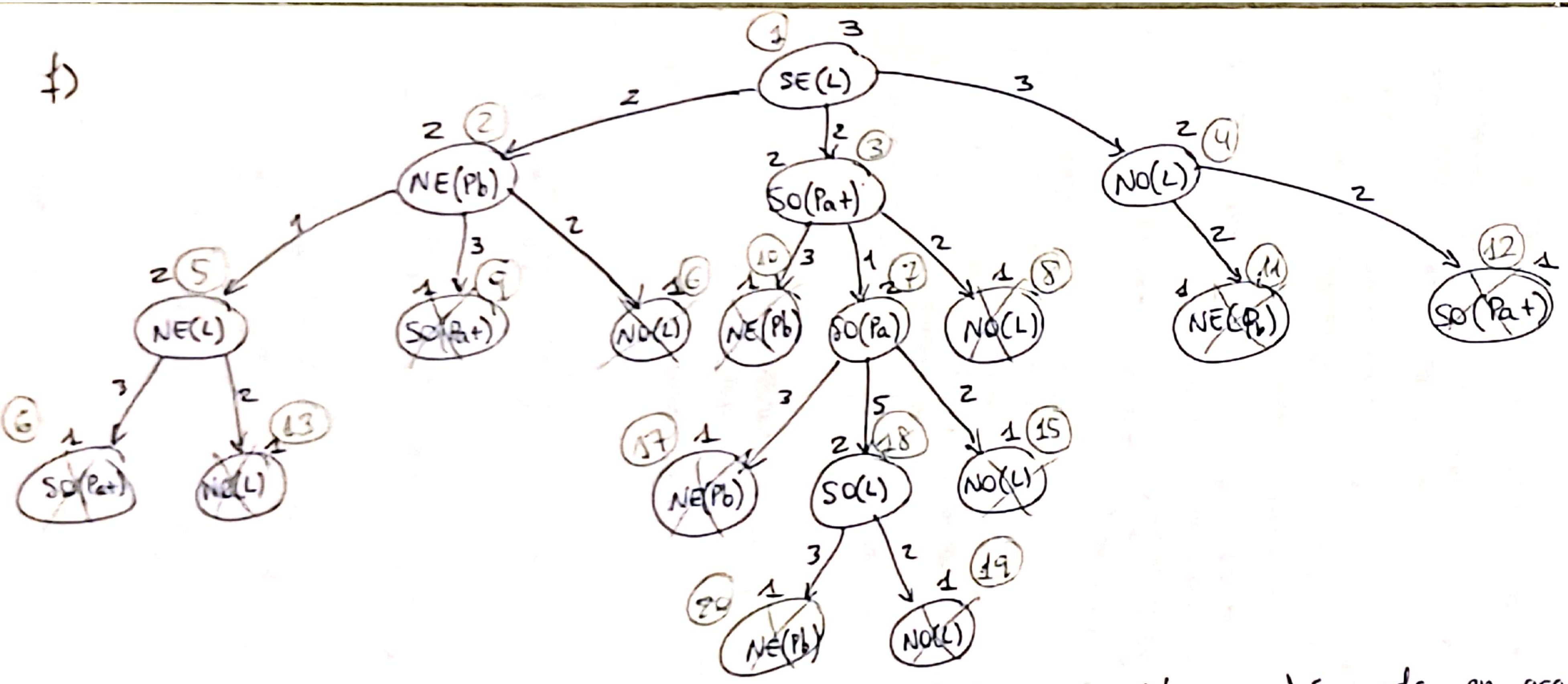
demostración admisibilidad:

coste máximo $= 4$

coste mínimo para pasar por todos los sectores sin incluso limpiar $= 6$

$\Rightarrow$ Nunca sobreestima

f)

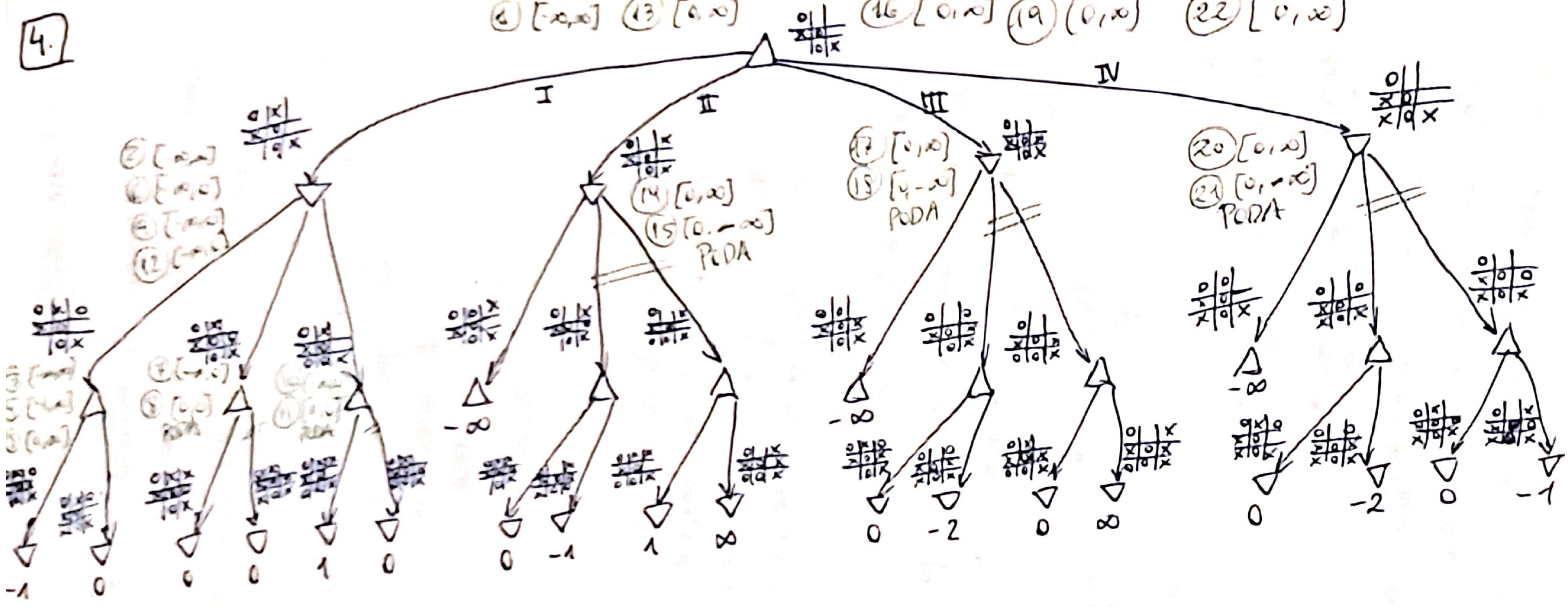


g) Según la estratégia usada, como A* con heurística admisible y búsqueda en grafo no es completo, no garantiza encontrar la solución, como es en este caso, debido a la eliminación de estados repetidos.

h) No

i) No. Hemos finalizado con que no hay solución, pero es trivial ver que sí que existe sin eliminar estados.

4.



Valor minimax raíz = 0 (empate)

Mejor jugada MAX = I

$\boxed{5.}$ a) FORMALIZACIÓN

i) Estados de búsqueda: cada estado se puede identificar como la tupla (a,b) con $a :=$ peso brazo izdo. y $b :=$ peso brazo dcho.

ii) Estado inicial: $(3,0)$ ó $(0,3)$ ambos válidos pero simétricos a efectos prácticos.

iii) Test objetivo: alcanzado estado final si $a = b$

```
bool goal_test_func (current_state)
    if current_state[a] == current_state[b]:
        TRUE
    else
        FALSE
```

iv) Acciones:

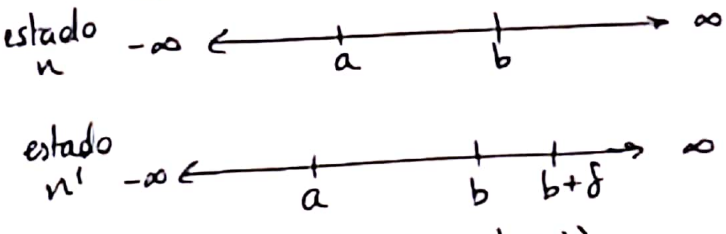
$4I := +4Kg$ brazo izdo, e.d. $4I := (a,b) \rightsquigarrow (a+4, b)$
$5I := +5Kg$ brazo izdo., e.d. $5I := (a,b) \rightsquigarrow (a+5, b)$
$4D := +4Kg$ brazo dcho., e.d. $4D := (a,b) \longrightarrow (a, b+4)$
$5D := +5Kg$ brazo dcho., e.d. $5D := (a,b) \rightsquigarrow (a, b+5)$

b) $h(n) := abs(a-b) \equiv |a-b|$

Es una heurística monótona por lo siguiente:

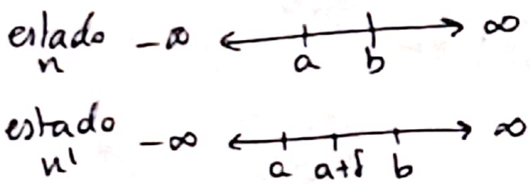
$$0 \leq h(n) \leq \underbrace{\Gamma_{n \to n'}}_{\in \{4,5\} = \delta} + \underbrace{h(n')}_{\geq 0} \quad n' \text{ sucesor } n$$

situación 1:



entonces $h(n) < h(n')$

situación 2:

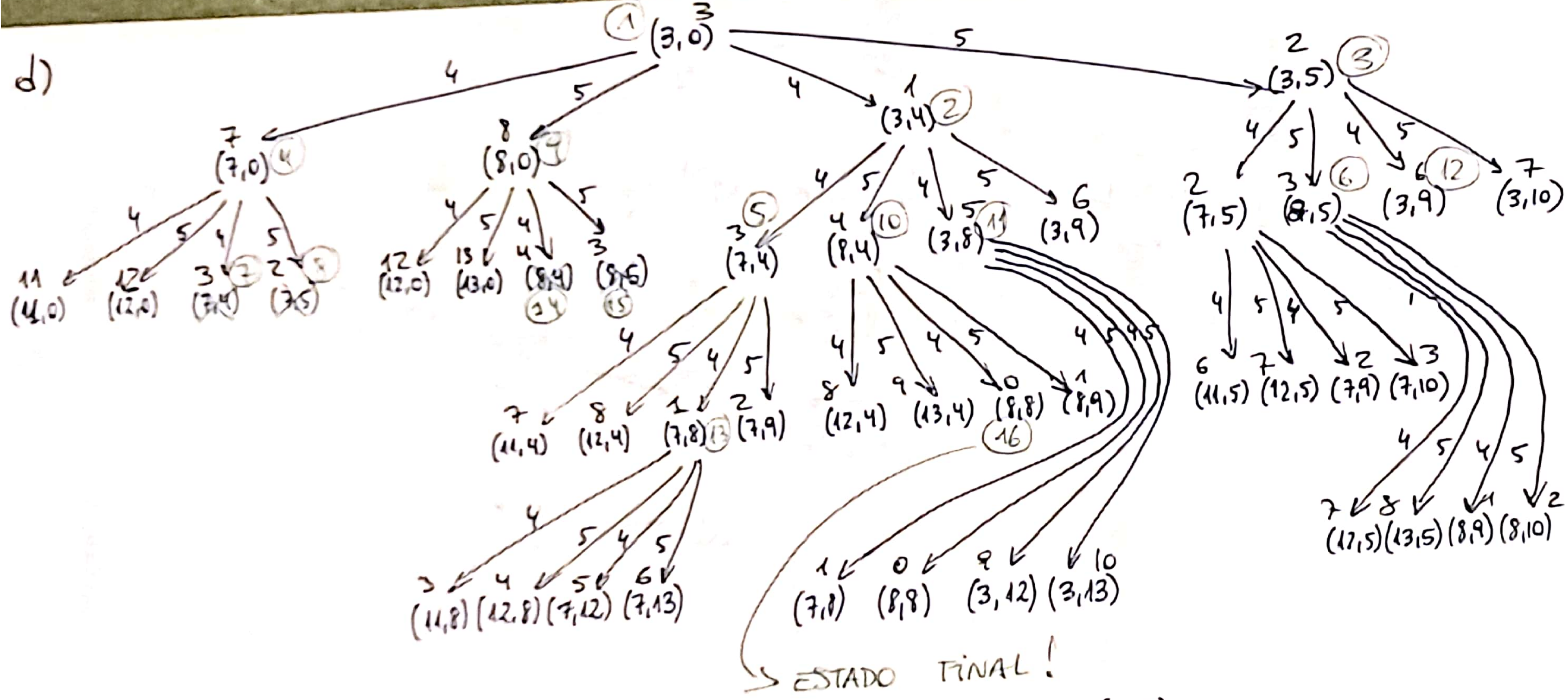


entonces $h(n) = h(n') + \Gamma_{n \to n'}$ porque la distancia mejorada de $h(n)$ con $h(n')$ es el valor de $\Gamma_{n \to n'}$.

Análogo para los casos $a \geq b$ (a a la dcha. de b en la recta real).

c) Monótona $\Longrightarrow$ Admisible.

d)



e) Solución encontrada $(3,0) \xrightarrow[4D]{} (3,4) \xrightarrow[5I]{} (8,4) \xrightarrow[4D]{} (8,8)$

COSTE $= 4+5+4 = \underline{\underline{13}}$

f) Como es A^* con búsqueda en grafo + heurística monótona $\Rightarrow$ óptima + completa $\Rightarrow$

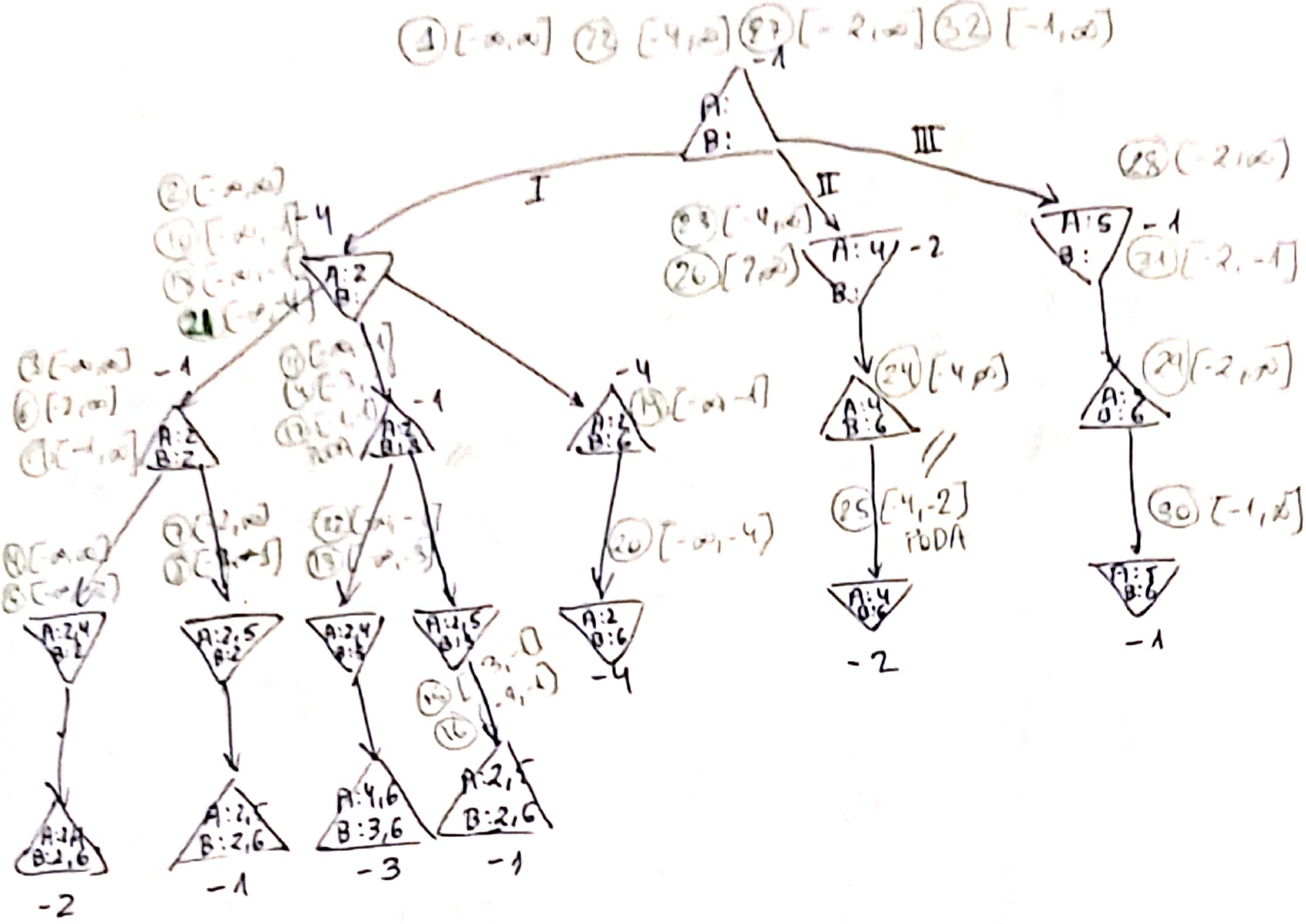
$\Rightarrow$ Sí que es una solución óptima.

6.

i)

a) Determinista b) Completa c) Suma cero

ii) Minimax

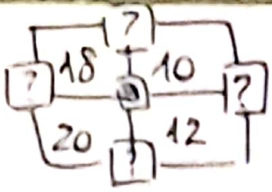
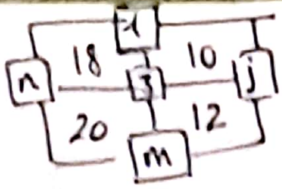


iv) Minimax

a) -1
b) III
c) A juega 5, B solo puede jugar 6, fin partida
d) B

v) Todo igual

7

(i) a) Estado inicial  = 

b) Test objetivo.

Consiste en comprobar que $i+j+3=10$, $m+i+3=12$, $n+m+3=20$ y $n+i+3=18$.

```
bool test-objetivo(estado-puzle).
    if j+i+3 ≠ 10:
        FALSE
    if j+m+3 ≠ 12:
        FALSE
    if m+n+3 ≠ 20.
        FALSE
    if n+i+3 ≠ 18:
        FALSE
    TRUE
```

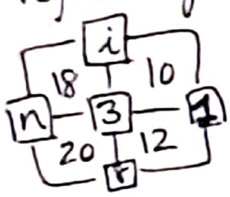
c) Acción 1: escoger dos índices adyacentes de $\{i,j,m,n\}$ y darles valores tales que cumplan su respectiva ecuación. Como son dos incógnitas para una ecuación existen varias posibilidades.

Ejemplo: escogemos j, m y le damos dos valores de $\{1,2,4,5,6,7,8,9\}$ (no hay repeticiones) tal que $j+m+3=12$.

Acción 2: Completar un cuadrado negro tal que su suma con los contiguos de el valor del cuadrado blanco.

Siguiendo el ejemplo anterior, si $\underbrace{j=1,\ m=8}_{\text{coste } 9}$, entonces $i+1+3=10 \rightarrow \underbrace{i=6}_{\text{coste } 6}$

(ii) Heurística monótona: suma de las diferencias entre la suma de los cuadrados negros y su respectivo cuadrado blanco.

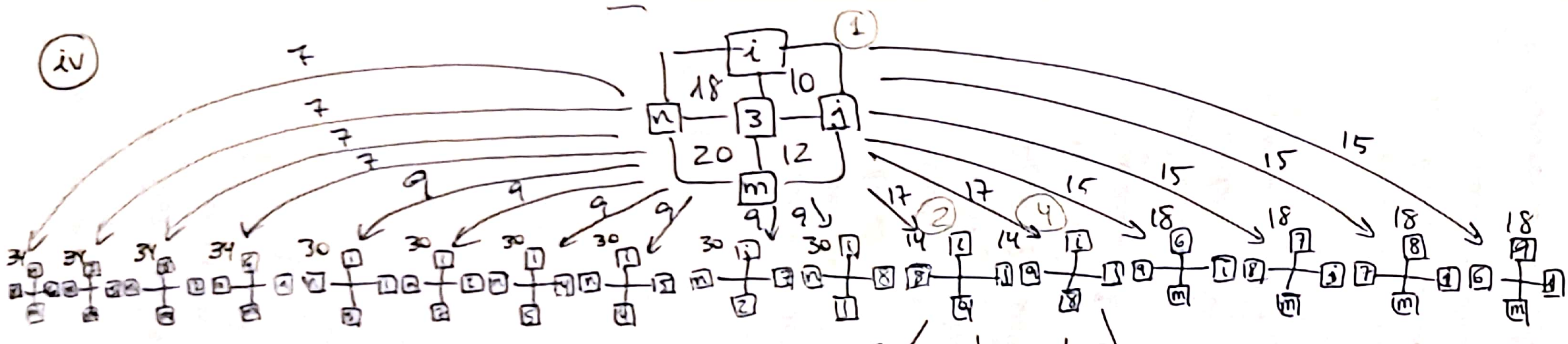
Ejemplo:  $h(n) = (10-(3+1)) + (12-(3+1+8)) + (18-3) + (20-(3+8))$

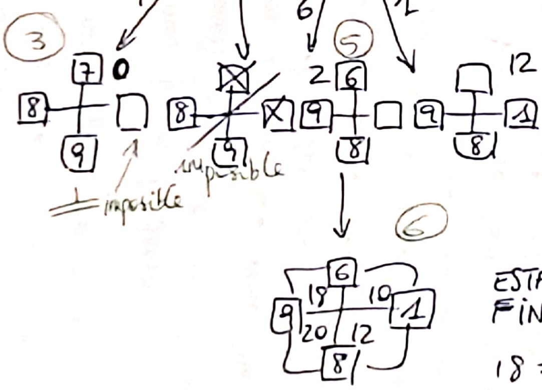
$= 6+0+15+9=30$

Es monótona porque el coste de una acción es igual a anular un término de la suma de diferencias entre la suma de los cuadrados negros y su blanco. $h(n) = \Gamma_{n \to n'} + h(n')$ n' sucesor n.

(iii) Monótona $\Rightarrow$ Admisible

(iv)





(v) Sí

A^* con búsqueda en grafo y usando una heurística monótona es completa y óptima.

ESTADO FINAL

$18 = 6+3+9$

$10 = 6+1+3$

$12 = 1+8+3$

$20 = 8+9+3$

$\boxed{8}$

(i) ¿ $h(A) \leq \Gamma_{A \to B} + h(B)$? $\longleftrightarrow$ ¿ $4 \leq 3 + 2$? $\Rightarrow$ NO $\Rightarrow$

$\Rightarrow$ $h(n)$ NO es MONÓTONA

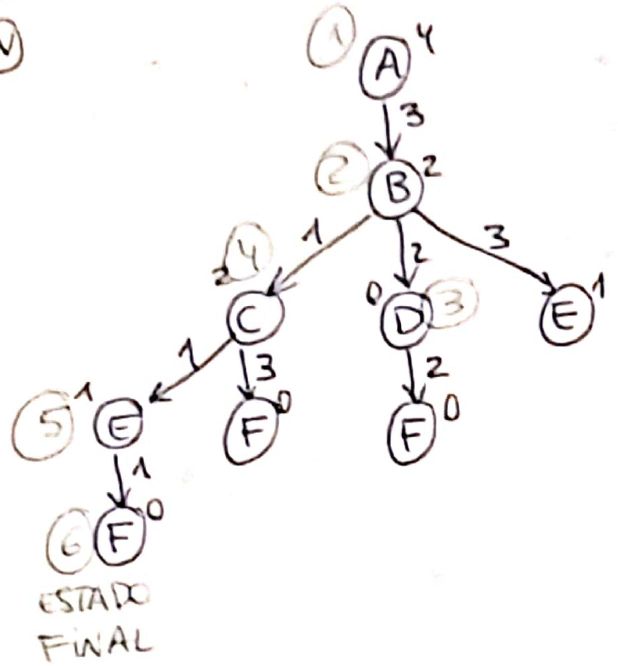
(ii) Sí, porque $h(n) \leq \underline{h^*(n)}$ $\hookrightarrow$ coste real

Esto se puede comprobar nodo por nodo observando que $h(n) \leq$
$\leq \Gamma_{n \to n'} + \cdots + \Gamma_{n'' \to F}$ $\forall n$ y n' sucesor y n'' sucesor de sucesores

(iii) Sí (Teorema)

(iv) No (Teorema)

(v)



(vi) Solución obtenida $A \to B \to C \to E \to F$ con coste 6 (óptimo). Resulta ser la solución óptima porque no hemos tenido que eliminar ningún estado.

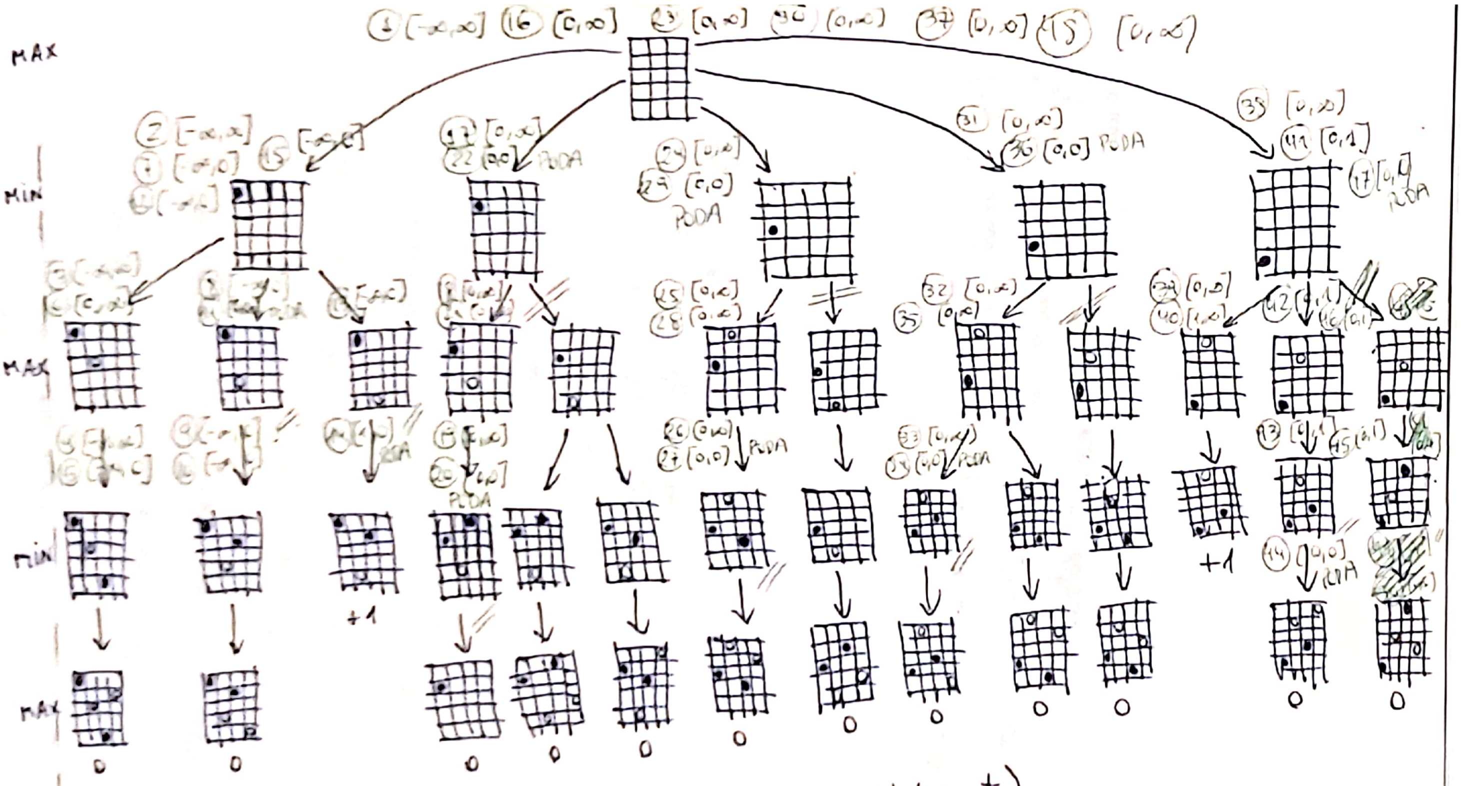
9.

i) a) Determinista

b) Completa

c) Suma constante

ii) Minimax



MAX (A) siempre pierde (si MIN (B) es inteligente)