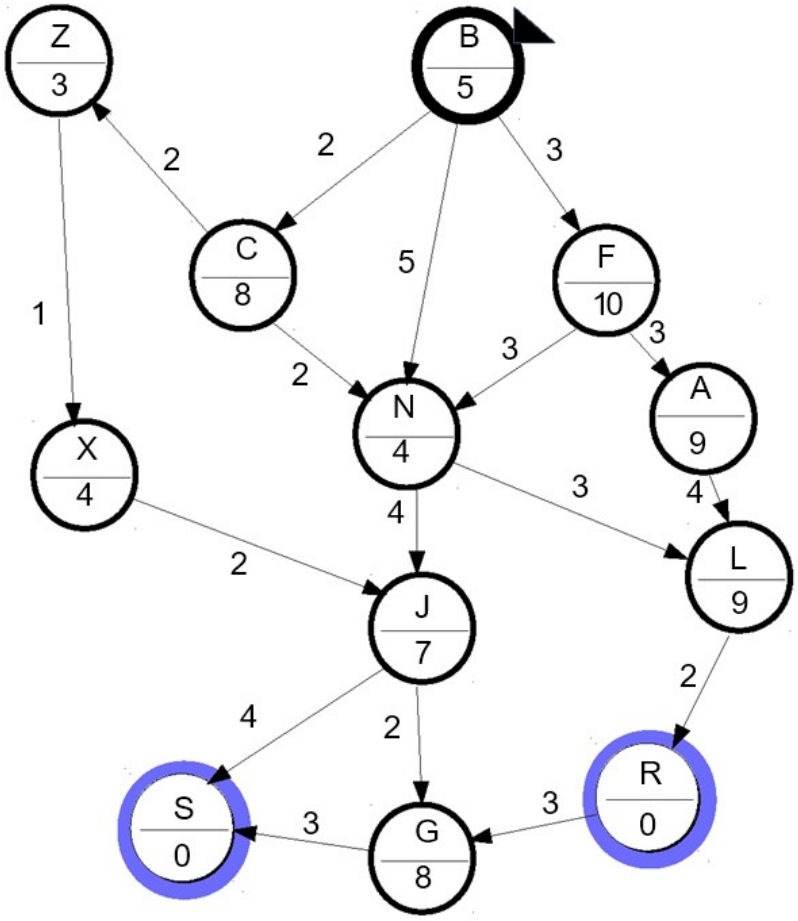
Inteligencia Artificial
Departamento de Ciencias de la Computación e Inteligencia Artificial
Curso 2019/2020 (Relación de Problemas Entregable)

Nombre:	Blanca Cano Camarero		
DNI:	75577392J		
Grupo:	DGIIM	Subgrupo Prácticas	IA3

Se deberá subir una copia de este documento completada y en formato PDF en DECSAI desde el 16 de junio a partir de las 10:00 y con fecha límite el 18 de junio a las 10:00.

Ejercicio 1: *(6 Puntos)* Supongamos que el siguiente grafo representa un espacio de estados de un problema real. El número dentro de cada nodo representa el valor de la función ***h*** del nodo y el número sobre el arco el coste del operador. El único nodo de inicio es ***B***. Los nodos objetivos son ***R*** y ***S***.

C



Usar los algoritmos de búsqueda especificados más abajo para resolver el problema anterior. En todos los casos ABIERTOS representará el conjunto de nodos que pueden ser explorados en la siguiente iteración del algoritmo de búsqueda. Y por supuesto, como en todos los algoritmos de búsqueda, el algoritmo termina cuando un nodo objetivo satisface la condición de terminación del propio algoritmo, y en ningún caso es necesario obtener todos los nodos objetivos.

Importante: En la búsqueda en anchura los nodos se deben expandir siguiendo el orden alfabético (por ejemplo, los sucesores del nodo B son L, M y Z en ese orden). En los otros dos algoritmos cuando se presente alguna situación de empate se aplicará de nuevo el orden alfabético para seleccionar los nodos. Esto es fundamental para que el ejercicio se pueda valorar correctamente, de no usar el criterio, los problemas no estarán bien resueltos.

Para cada algoritmo se pedirán cuatro datos:

- Secuencia de nodos: corresponde a la secuencia de nodos en el orden en el que salen de
- ABIERTOS.
- El objetivo alcanzado.
- El camino solución encontrado: secuencia de nodos del camino empezando en B.
- El coste de la solución encontrada. En el algoritmo de búsqueda en anchura no se usa el coste, pero el camino final tiene un coste, reflejarlo en la casilla correspondiente. Los otros dos algoritmos sí lo usan.

Algoritmo 1: Búsqueda en Anchura **(1 punto)**

Secuencia de nodos	• B C F N Z A J S (si antes de meterlo en abiertos comprueba que es solución). • B C F N Z A J L X G S (Si comprueba si el nodo es solución al sacarlo de abierto)
Estado objetivo alcanzado	S
Camino solución	B N J S
Coste de la solución	13

Insertar aquí documentación gráfica de la resolución manual:

Anchura

Notación el conjunto A hace referencia a los nodos abiertos, C lo hará para los cerrados.

El par (x,y) hace referencia al nodo x, cuyo nodo sucesor es y

1. $A = \{(B, \emptyset)\}$ $\quad C = \emptyset$; como B no es el objetivo expandimos el nodo B y ordenamos según criterio alfabético.

2. $A = \{(G, B), (F, B), (N, B)\}$ $\quad C = \{(B, \emptyset)\}$

El nodo G no es el objetivo: lo expandimos.

Notemos que N NO se vuelve a meter por estar repetido.

3. $A = \{(F, B), (N, B), (K, C)\}$ $\quad C = \{(B, \emptyset), (G, B)\}$

(K, C) Nodo expandido

Repetimos proceso anterior, reordenando nodos del mismo nivel.

Tras expandir (F, B)

4. $A = \{(N, B)\ (K, C), (A, F)\}$ $\quad C = \{(B, \emptyset), (G, B), (F, B)\}$

Misma situación tras expandir (N, B)

5. $A = \{(K, C), (A, F), (J, N), (L, N)$

$C = \{(B, \emptyset), (G, B), (F, B), (N, B)\}$

K No es solución, expandimos el nodo.

6. $A = \{(A,F), (J,N), (L,N), (X,Z)\}$

$C = \{(B,\emptyset), (C,B), (F,B), (N,B), (Z,C)\}$

7. A no es nodo objetivo

$A = \{(J,N), (L,N), (X,Z)\}$

$C = \{(B,\emptyset), (C,B), (F,B), (N,B), (Z,C), (A,F)\}$

$A = \{(L,N), (X,Z), (G,J), (S,J)\}$

Puesto que estamos en una cola, el primer nodo objetivo en salir será S. (tras repetir el algoritmo 4 veces).

Ahora nos queda reconstruir la secuencia de antecesores fijándonos en la lista de cerrados.

El antecesor de S es J, el de J es N y para este último B; la raíz.

Luego la secuencia es: B, N, J, S.

La suma de los arcos que unen estos nodos es $5+4+4=13$

Por tanto el costo resulta 13.

Algoritmo 2: Búsqueda Costo Uniforme **(2 puntos)**

Secuencia de nodos	B C F N Z X A J L G R
Estado objetivo alcanzado	R
Camino solución	B C N L R
Coste de la solución	9

Costo uniforme

(X, Y, C) x Nodo, Y nodo mejor padre, c costo

Iter abierto	Nodo	Iter cerrado	Mejora	Iters mejor
0	(B, ∅, 0)	1		
1	(C, B, 2)	2		
1	(N, B, 5)	4	(N, C, 4)	2
1	(F, B, 3)	3		
2	(Z, C, 4)	5		
3	(A, F, 6)	7		
4	(J, N, 8)	8	(J, X, 7)	5
4	(L, N, 7)	9		
5	(X, Z, 5)	6		
8	(S, J, 11)			
8	(G, J, 9)	10		
9	(R, L, 9)			

Cerrados

1 (B, ∅, 0)
2 (C, B, 2)
3 (F, B, 3)
4 (N, C, 4)
5 (Z, C, 4)
6 (X, Z, 5)
7 (A, F, 6)
8 (J, X, 7)
9 (L, N, 7)
10 (G, J, 9)
FIN (R, L, 9)

Resumen resultados

- Objetivo alcanzado: R
- Secuencia inversa: R, L, N, C, B
- Camino solución: B C N L R
- Costo: 9

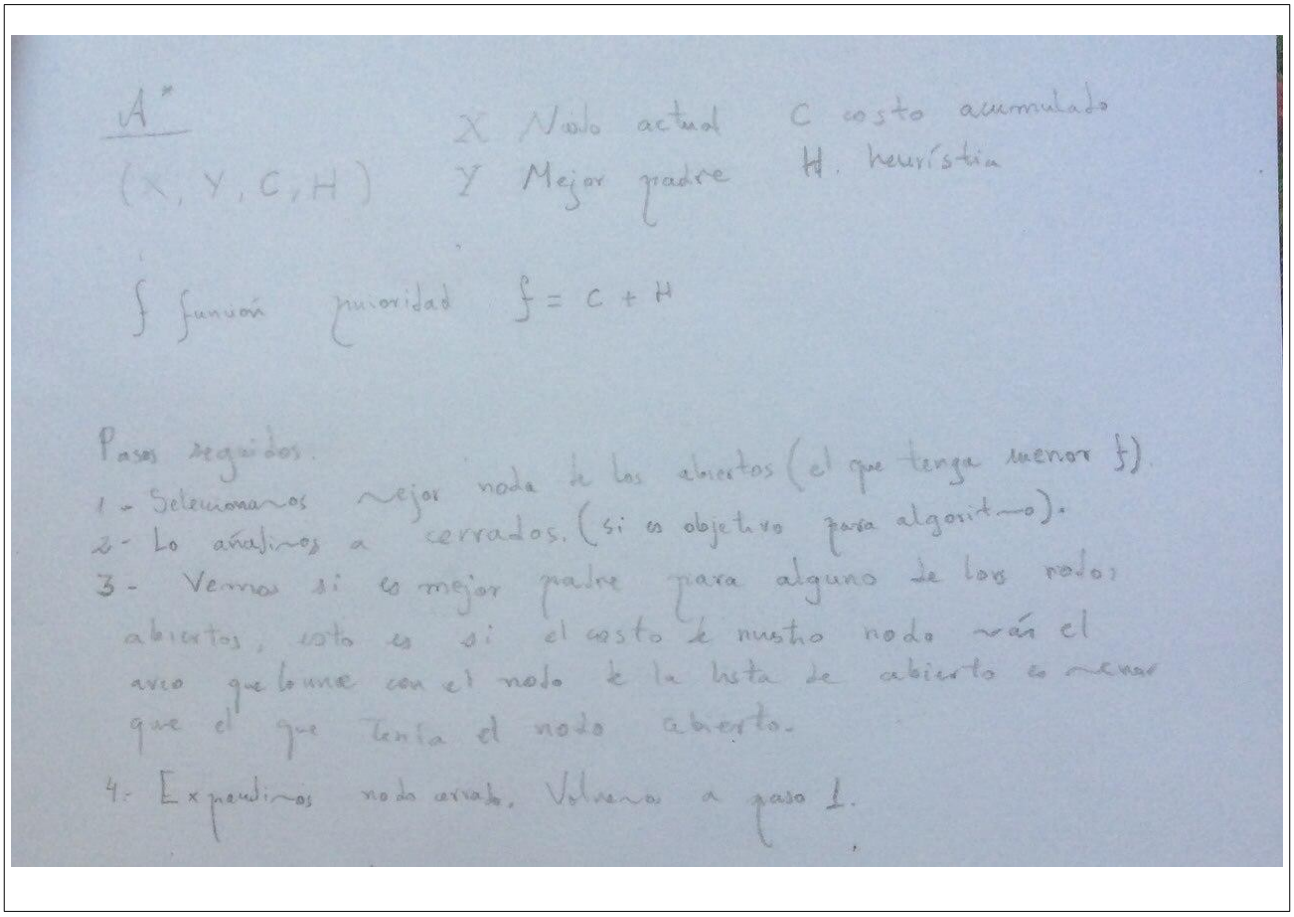
Descripción del algoritmo

1. Sacar de abiertos el nodo de menor costo (en caso de igualdad aplicar criterio alfabético).
2. Si N es solución FIN. Si no expandir añadir nuevos nodos a abierto y si mejora costo de alguno actualizar.
3. Añadir N a cerrados, volver paso 1

Algoritmo 3: Búsqueda A* **(3 puntos)**

Secuencia de nodos	B N C Z X F J S
Estado objetivo alcanzado	S
Camino solución	B C Z X J S
Coste de la solución	11

Insertar aquí documentación gráfica de la resolución manual:



Iter aparición	Iter cerrados	Nodo	f	iter mejora	Nuevo Nodo	Iter mejora	Nodo mejora 2
0	1	(B, ∅, 0, 5)	5				
1	3	(C, B, 2, 8)	10				
1	2	(N, B, 5, 4)	9				
1	6	(F, B, 3, 10)	13				
2	7	(J, N, 9, 7)	16 16 14	3	(J, N, 8, 7)	5	(J, X, 7, 7)
2		(L, N, 8, 9)	~~16~~	3	(L, N, 7, 9)		
3	4	(Z, C, 4, 3)	7				
4	5	(X, Z, 5, 4)	9				
6		(A, F, 6, 9)	15				
7		(S, J, 11, 0)	11				
7		(G, J, 9, 8)	14				

Cerrados

1 (B, ∅, 0, 5)
2 (N, B, 5 4) → (iter 3) (N, C, 4, 4)
3 (C, B, 2, 8)
4 (Z, C, 4, 3)
5 (X, Z, 5, 4
6 (F, B, 3, 10)
7 (J, X, 7, 7)

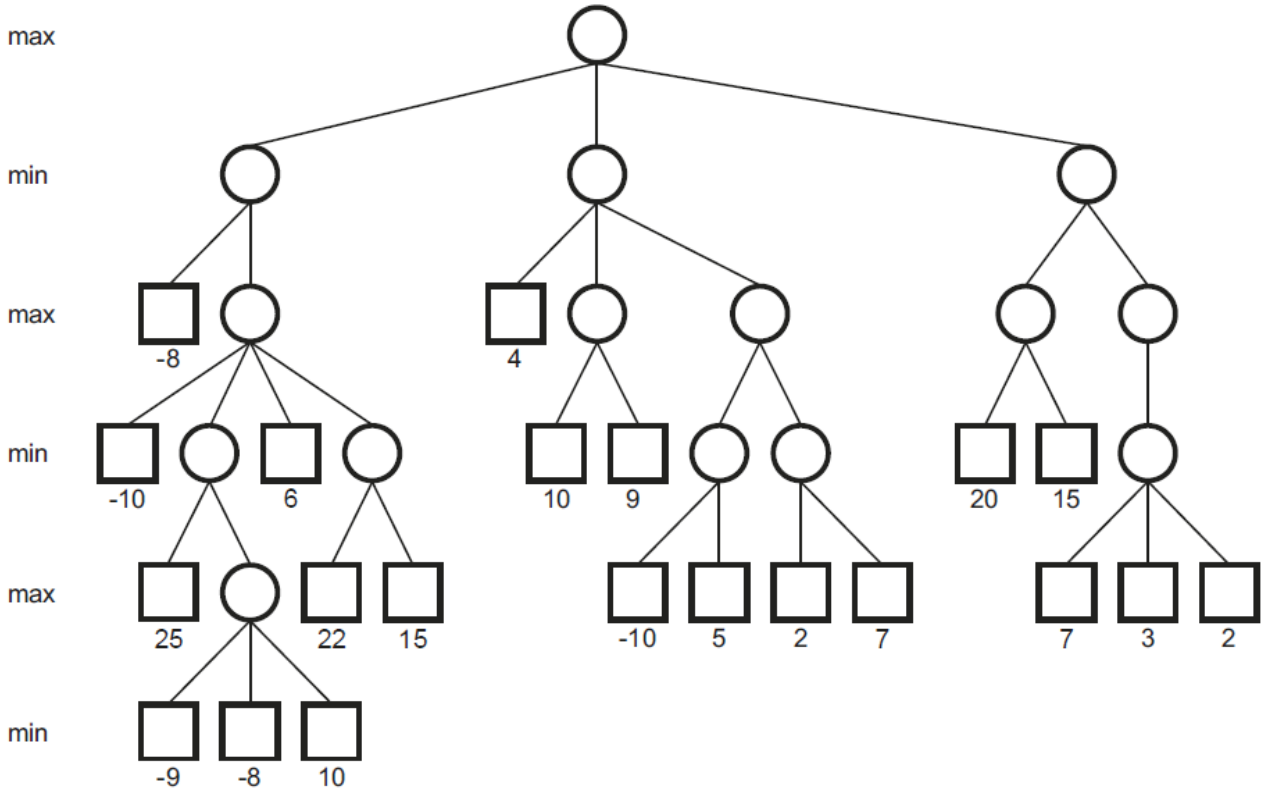
F. n8 (S J 11, 0)

Resumen soluciones

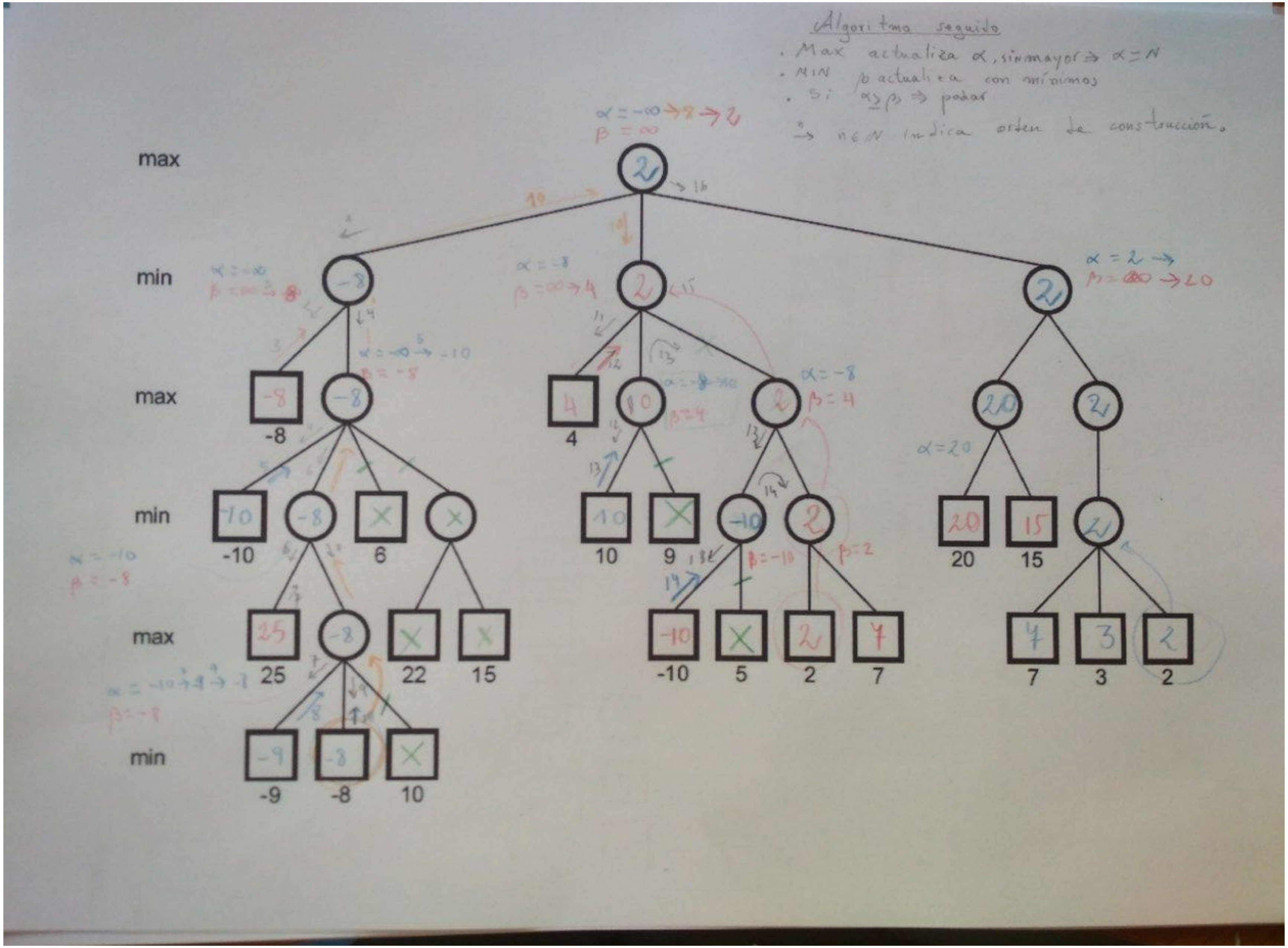
- Objetivo S
- Ruta inversa: S, J, X, Z, C, B
- Camino solución: B, C, Z, X, J, S
- Costo: 11

Ejercicio 2: *(4 puntos)* En la siguiente página se muestra un árbol de búsqueda para un determinado juego. Las capas de nodos MAX y MIN están especificadas en el lateral izquierdo de la imagen. Los nodos en forma de circulo representan los nodos interiores del árbol. Los nodos en forma de rectángulo representan estados terminales del juego y en los que se debe aplicar la función de evaluación estática o heurística. El valor heurístico asociado a cada nodo terminal se indica debajo de dicho nodo.

Se pide resolver el juego usando el algoritmo de poda alfa-beta. Es muy importante que en su resolución aparezcan claramente marcados los nodos terminales podados (con forma de rectángulo) **(marcar con una "X" el interior del recuadro de aquellos nodos que no se evaluarán mediante su función heurística)**, así como **el valor Minimax** asociado al nodo de inicio, indicar también las **podas que se hayan producido**, si hubiese alguna, marcando con una cruz sobre los arcos podados.



Indicar aquí el valor minimax obtenido=2


Algoritmo seguido
· Max actualiza α, si mayor ⇒ α = N
· MIN β actualiza con mínimos
· Si α ≥ β ⇒ podar
→ n ∈ N indica orden de construcción.
max
min
max
min
max
min
-8
4
-10
6
10
9
20
15
25
22
15
-10
5
2
7
7
3
2
-9
-8
10