# Sistemas Inteligentes – Examen Bloque 1, 4 noviembre 2022
# Test B (1,75 puntos) puntuación: max (0, (aciertos – errores/3)*1,75/9)

| | |
|---|---|
| **Apellidos**: | **Nombre**: |
| **Grupo**: A B C D E F G 4IA | |

1) Dado el SBR, indica la afirmación **CORRECTA** tras realizarse el primer *pattern-matching*:

```
(deffacts prueba
     (prueba 1 2 3 4 5 6 7 8 9 10))

(defrule regla
     ?f1 <- (prueba $?a ?b $?c ?d)
=>
     (retract ?f1)
     (assert (lista $?c)))
```

A. Se producirán 11 instanciaciones
B. Se producirán 10 instanciaciones
C. Se producirán 9 instanciaciones
D. No se producirá ninguna instanciación

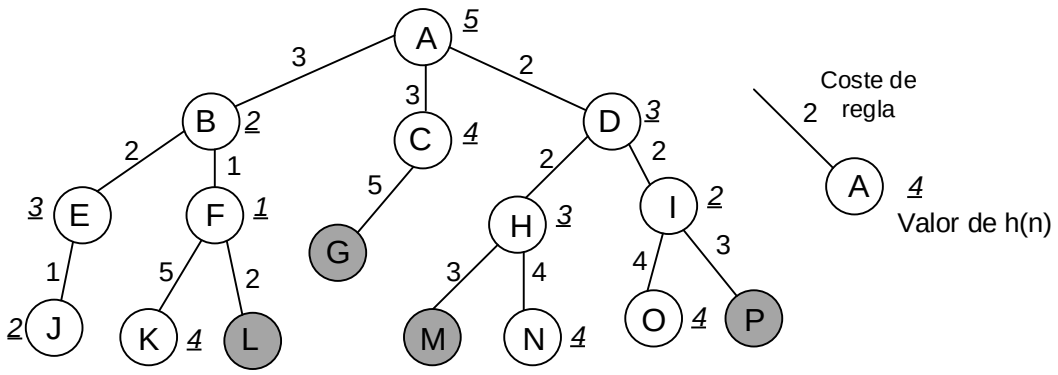
---

2) Dados tres algoritmos A para un mismo problema, donde A1 y A2 usan una heurística admisible, tal que ∀n, h*(n) ≥ h2(n) > h1(n). La heurística h3(n) del algoritmo A3 no es admisible. Se puede asegurar que:

A. Los tres algoritmos A podrían encontrar la misma solución.
B. El algoritmo A1 generará menos nodos que A2.
C. La solución encontrada por A1 será mejor que la encontrada por A3.
D. El algoritmo A3 generará más nodos que A2.

---

3) Asumiendo que aplicamos diferentes estrategias de búsqueda al espacio de estados de la figura, donde a igualdad de criterio, se elige el nodo alfabéticamente menor, indica cuál de las siguientes afirmaciones es **CORRECTA**:



A. La solución que encuentra una búsqueda de tipo A es el nodo P
B. Una estrategia de coste uniforme devolverá la misma solución que un algoritmo de tipo A
C. La función h(n) no es consistente (monótona)
D. La aplicación de un algoritmo en anchura devuelve la solución óptima

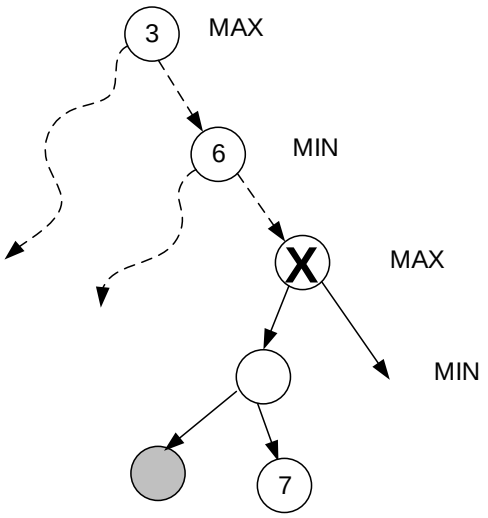
---

4) Dada la siguiente base de hechos BH={(ciudades Valencia min 14 max 36 Alicante min 18 max 34 Castellón min 12 max 30 Requena min 8 max 38)(selección Castellón)}, se desea hacer una regla que calcule la diferencia de temperatura (máxima – mínima) de la ciudad seleccionada. ¿Cuál de los siguientes patrones **NO SERVIRÍA** para la parte izquierda de la regla?

A. (selección ?c) (ciudades $?x ?c min ?t1 max ?t2 $?w)
B. (selección ?c) (ciudades $?x ?c ? ?t1 ? ?t2 $?w)
C. (selección ?c) (ciudades $? ?c ?n1 ?t1 ?n2 ?t2 $?)
D. (selección ?c) (ciudades $?x ?c ? ?t1 ? ?t2 $?x)

---

5) ¿Qué valor debería tener el nodo sombreado para que se produzca el corte indicado?




A. Cualquier valor comprendido en [3, ∝]
B. Cualquier valor comprendido en [6, ∝,]
C. Cualquier valor comprendido en [-∝, 6]
D. Cualquier valor comprendido en [-∝, 7]

---

6) En una búsqueda en Profundidad, con máximo nivel de profundidad **m>=1**, se encuentra una solución en el nivel **m**. Sabiendo que no existe ninguna solución en un nivel menor que **m**, y que la solución encontrada es la única que existe a nivel **m**, indica la respuesta **INCORRECTA**:

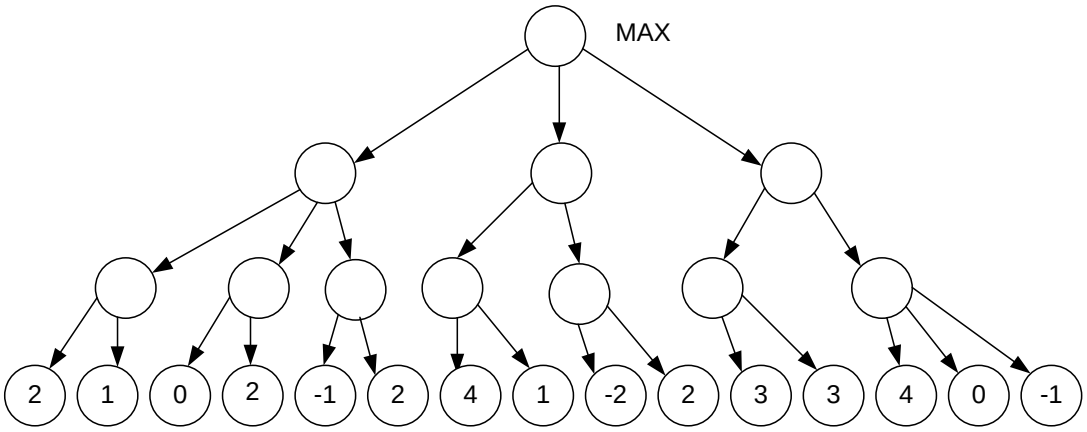
A. Una estrategia de Anchura encontrará la misma solución que Profundidad
B. El número de nodos que generará Anchura será siempre mayor que en Profundidad
C. Una estrategia de Profundización Iterativa encontrará la misma solución que Profundidad
D. El número de nodos que generará Profundización Iterativa será siempre mayor que en Profundidad

7) Se desea realizar una búsqueda en CLIPS. Para ello, las reglas no deben contener la instrucción *retract* porque:

A. No permitiría explorar caminos alternativos
B. No permite calcular el valor de g(n)
C. No permite encontrar la solución óptima
D. Ninguna de las anteriores

---

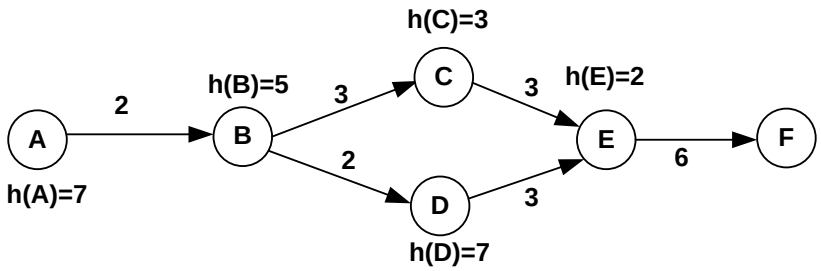
8) Dado el árbol de juego de la figura, donde se aplica un procedimiento alfa-beta, indica cuántos nodos evitamos generar respecto a un algoritmo MINIMAX si realizamos una exploración alfa-beta:




A. 2
B. 4
C. Más de 4 y menos de 7
D. Más de 6

---

9) Dado el espacio de estados de la figura, donde A es el estado inicial, F el nodo meta, y se indican los costes de cada arco y la estimación h(n) de cada nodo, marca la opción **CORRECTA**:




A. La aplicación de un algoritmo A (GRAPH SEARCH, con control de nodos repetidos en CLOSED, descartando directamente el nodo si ya existe en CLOSED) obtendrá la senda óptima.
B. La aplicación de un algoritmo A (TREE SEARCH, con control de nodos repetidos en OPEN) no obtendrá la senda óptima.
C. La respuesta A no es cierta debido a que h(n) no es consistente
D. La respuesta B no es cierta, debido a que h(n) no es admisible

# Sistemas Inteligentes – Examen Bloque 1, 4 noviembre 2022
# Problema: 2 puntos

La compañía EuroTransport se dedica al transporte aéreo de mercancías entre 6 ciudades europeas: Valencia, Barcelona, París, Roma, Berlín y Oslo. La compañía dispone de varios aviones ubicados en las 6 ciudades europeas mencionadas y realiza una planificación diaria de las rutas que realizarán los aviones. La ruta de cada avión es fija y contiene un conjunto de ciudades que incluye la ciudad final de destino y posibles ciudades intermedias donde el avión hace escala. Por ejemplo, la ruta de un avión ubicado en Valencia puede ser Valencia Berlín; la ruta de otro avión ubicado en Barcelona puede ser Barcelona París Oslo.

Existe un conjunto de paquetes a transportar que están ubicados en alguna de las ciudades y cuyo destino es otra de las ciudades (por ejemplo, un paquete está en París y su destino es Oslo). Un paquete se puede cargar en un avión si la ruta del avión tiene una parada en la ciudad destino del paquete, bien sea como escala intermedia o como destino final.

El patrón que debe seguirse necesariamente para representar la información dinámica de este problema es:

$$(\texttt{eurotransport}\ [\texttt{avion num\_a}^{s}\ \texttt{c\_a}^{s}\ \texttt{p\_a}^{m}\ \texttt{finav}]^{m}\ [\texttt{paq c\_p}^{s}\ \texttt{cd\_p}^{s}]^{m})$$

donde c_a, p_a, c_p, cd_p ∈ {VAL, BAR, PAR, ROM, BER, OSL} y num_a ∈ INTEGER, siendo:

num_a número del avión
c_a ciudad donde se encuentra el avión
p_a ciudad destino de los paquetes cargados en el avión (los paquetes se representan mediante una etiqueta que es su ciudad destino; por ejemplo, si solo hay dos paquetes con destino Berlín en un avión, esta variable será BER BER )
c_p ciudad donde se encuentra el paquete
cd_p ciudad destino del paquete

Utilizaremos además el siguiente patrón para representar las rutas de los aviones:

$$(\texttt{ruta num\_a c\_t}^{m})$$ donde

num_a es el número del avión
c_t ∈ {VAL, BAR, PAR, ROM, BER, OSL} es la lista de ciudades que recorre el avión siendo la primera ciudad de la lista la ciudad origen del avión y la última la ciudad del destino final del avión

Usando CLIPS y búsqueda en grafos (GRAPH-SEARCH), se pide:

1) (0.4 puntos) Escribir la **base de hechos inicial** con la siguiente información: el avión 1 hace la ruta Valencia-Berlín; el avión 2 hace la ruta Barcelona-Roma-Berlín; el avión 3 hace la ruta Valencia-Barcelona-París-Oslo. Los aviones se sitúan inicialmente en la ciudad origen de su ruta. Hay tres paquetes a transportar: un paquete está en Barcelona con destino Oslo y otro paquete está en Roma con destino Berlín y el tercer paquete está en Valencia con destino Berlín. Inicialmente los aviones no contienen ningún paquete.

2) (0.5 puntos) Escribir una regla para mover un avión desde su actual ubicación a su siguiente destino según la ruta establecida, siempre y cuando el avión no se encuentre ya en su destino final. El resultado de ejecutar la regla es que el avión estará en su siguiente destino.

3) (0.7 puntos) Escribe una regla para cargar un paquete en un avión, el cual debe encontrarse en la misma ciudad que el paquete, siempre y cuando la ciudad destino del paquete aparezca en el resto de las ciudades a visitar en la ruta del avión. Como resultado de ejecutar la regla de cargar, el paquete (representado por la etiqueta de su ciudad destino) estará ahora dentro del avión y por tanto se elimina la información `[paq c_p^s cd_p^s]` del hecho.

4) (0.4 puntos) Escribe una regla que muestre por pantalla los aviones que han llegado a su destino final y no tienen paquetes. La regla deberá mostrar el mensaje "El avión XXX ha llegado a su destino final y no tiene paquetes" por cada uno de los aviones que cumplan las condiciones, donde XXX es el número del avión.

```
(deffacts datos
  (ruta 1 VAL BER)
  (ruta 2 BAR ROM BER)
  (ruta 3 VAL BAR PAR OSL)
  (eurotransport avion 1 VAL finav avion 2 BAR finav avion 3 VAL finav paq BAR OSL paq ROM BER paq VAL BER)
)


(defrule mover_avion
   (eurotransport $?x avion ?numa ?esta $?y)
   (ruta ?numa $?z ?esta ?destino $?w)
 =>
   (assert (eurotransport $?x avion ?numa ?destino $?y))
)


(defrule cargar_paquete_mismo_destino
   (eurotransport $?x avion ?numa ?esta $?y paq ?esta ?dest $?h)
   (ruta ?numa $? ?esta $? ?dest $?)
=>
   (assert (eurotransport $?x avion ?numa ?esta ?dest $?y $?h))
)


(defrule avion_en_destino
   (eurotransport $?x avion ?numa ?esta finav $?y)
   (ruta ?numa $? ?esta)
=>
   (printout t "El avion " ?numa " ha llegado a su destino final y no tiene paquetes " crlf)
)
```