

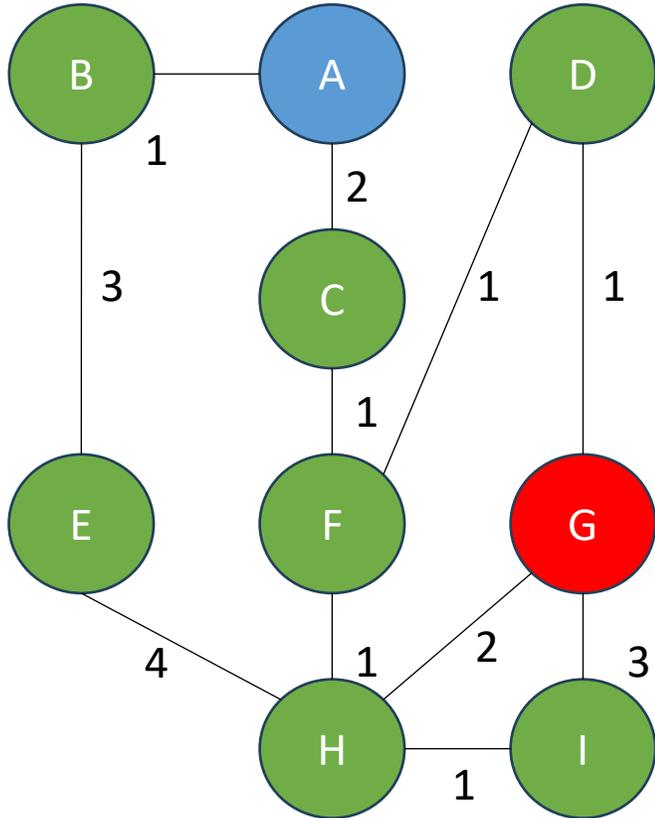
DESARROLLO DE JUEGOS CON INTELIGENCIA ARTIFICIAL

Búsqueda en el espacio de estados y agentes

Tema 2. Búsqueda no informada



Ejercicio 1



- Sobre el grafo anterior:
 - Siendo A el vértice de inicio.
 - Siendo G el vértice objetivo.
- Se pide aplicar:
 - BFS
 - DFS
 - UCS

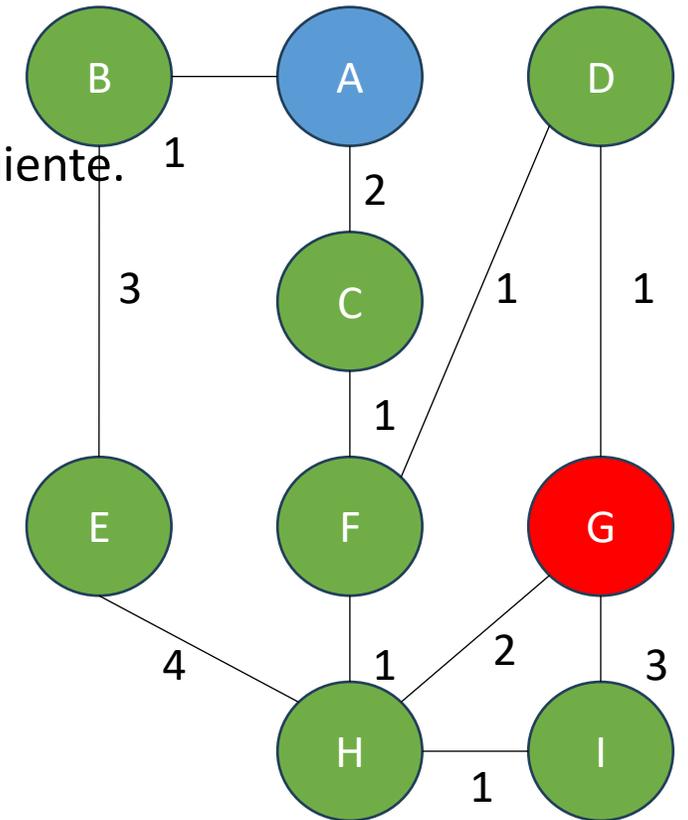
Ejercicio 1

- **BFS:**

- Comienza en el nodo A.
- Explora todos los nodos vecinos antes de avanzar a los nodos de nivel siguiente.
- Continúa hasta encontrar el nodo G.

- **Proceso BFS:**

- Paso 1: Visitar A y añadir B y C a la cola. Q: [B, C]
- Paso 2: Desencolar B y añadir E a la cola. Q: [C, E]
- Paso 3: Desencolar C y añadir F a la cola. Q: [E, F]
- Paso 4: Desencolar E y añadir H a la cola. Q: [F, H]
- Paso 5: Desencolar F (pero H ya está en la cola). Q: [H, D]
- Paso 6: Desencolar H y añadir G e I a la cola. Q: [D, G, I]
- Paso 7: Desencolar D y, no añaden nodos. Q: [G, I]
- Paso 8: Desencolar G. Aquí encontramos G. Q: [I]



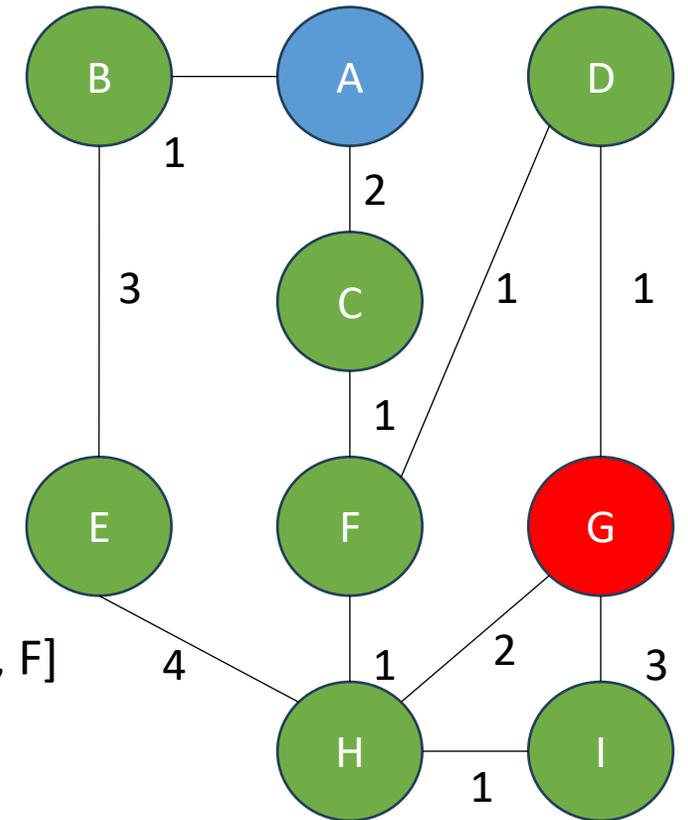
Ejercicio 1

- DFS:

- Comienza en el nodo A.
- Explora lo más profundamente posible antes de retroceder.
- Continúa hasta encontrar el nodo G.

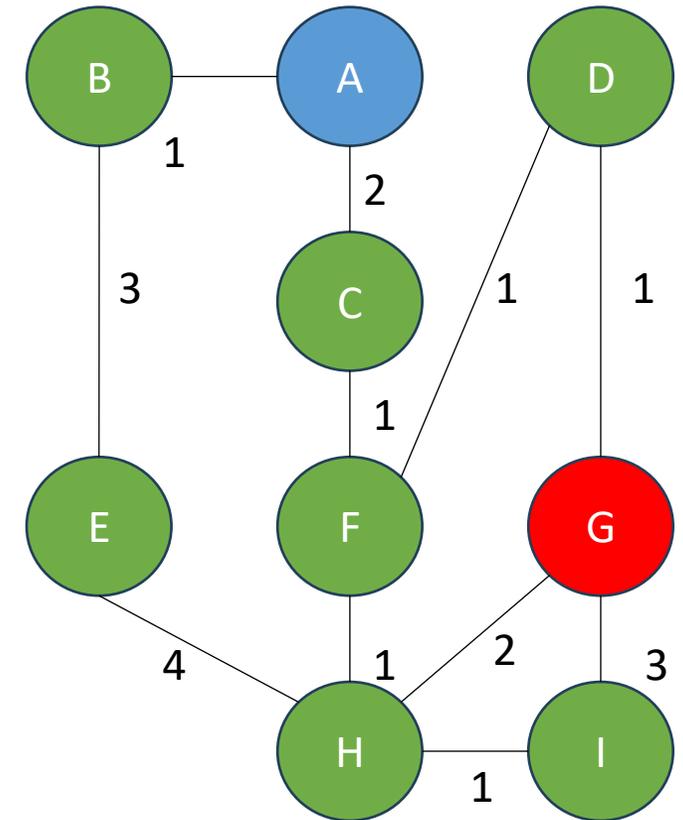
- Proceso DFS:

- Paso 1: Comenzamos en A. Añadimos C, y B a la pila. S: [C, B]
- Paso 2: Sacamos B de la pila y exploramos su sucesor E. S: [C, E]
- Paso 3: Sacamos E de la pila y exploramos su sucesor H. S: [C, H]
- Paso 4: Sacamos H de la pila y exploramos sus sucesores F, G e I. S: [C, I, G, F]
- Paso 5: Sacamos F de la pila y exploramos su sucesor D. S: [C, I, G, D]
- Paso 6: Sacamos D de la pila y, no se añade ningún nodo. S: [C, I, G]
- Paso 7: Sacamos G de la pila y, no se añade ningún nodo. S: [C, I]

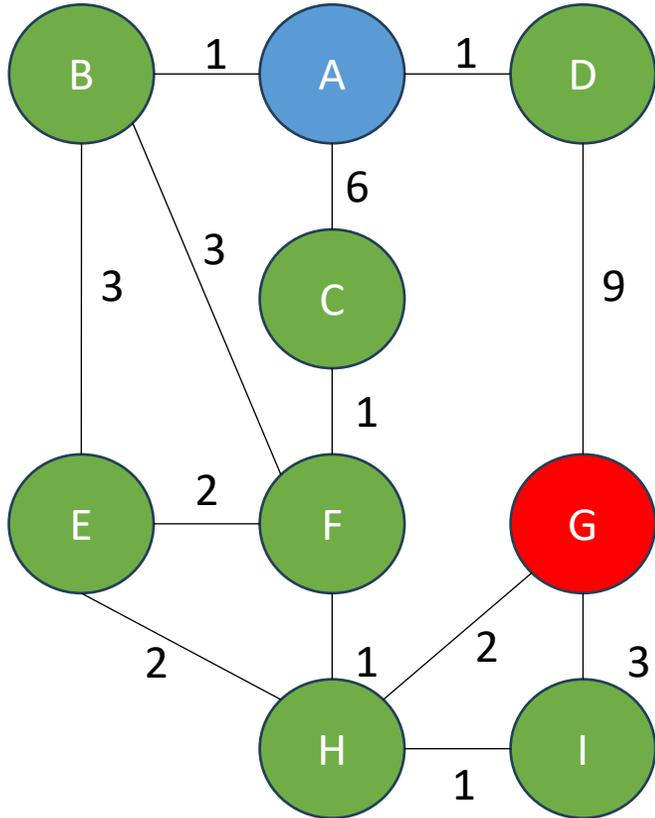


Ejercicio 1

- Paso 1: Inicialmente, solo el nodo A está en la cola. PQ: [(A, 0)]
- Paso 2: Expandimos A. Los sucesores B y C se añaden a la cola con sus respectivos costes acumulados.
PQ: [(B, 1), (C, 2)]
- Paso 3: Expandimos B (ya que tiene el coste acumulado más bajo). Se añade E con un coste acumulado de 4.
PQ: [(C, 2), (E, 4)]
- Paso 4: Expandimos C. Se añade F con un coste acumulado de 3. Cola: [(F, 3), (E, 4)]
- Paso 5: Expandimos F. Se añaden D y H con un coste acumulado de 4.
PQ: [(E, 4), (D,4), (H, 4)]
- Paso 6: Expandimos E. No se añaden nodos.
PQ: [(D, 4), (H, 4)]
- Paso 7: Expandimos D. Se añade G con un coste de 5.
PQ: [(H, 4), (G, 5)]
- Paso 8: Expandimos H. Se añade I con un coste de 6.
PQ: [(G, 5), (I, 6)]
- Paso 9: Expandimos G. Hemos llegado al objetivo. PQ [I, 6]
- Resultado UCS : A -> C -> F -> D -> G (coste Total: 5)



Ejercicio 2



- Sobre el grafo anterior:
 - Siendo A el vértice de inicio.
 - Siendo G el vértice objetivo.
- Se pide aplicar:
 - BFS
 - DFS
 - UCS

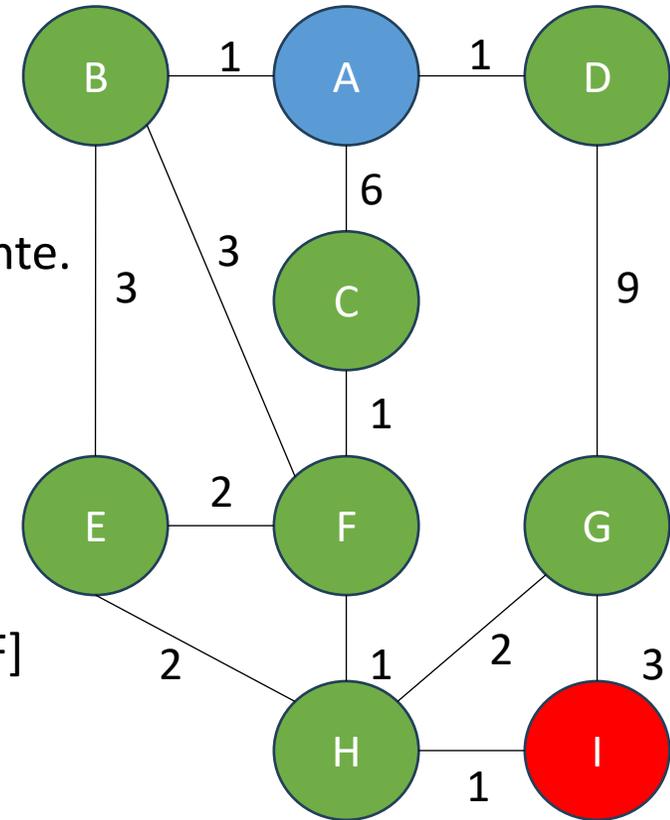
Ejercicio 2

- **BFS:**

- Comienza en el nodo A.
- Explora todos los nodos vecinos antes de avanzar a los nodos de nivel siguiente.
- Continúa hasta encontrar el nodo I.

- **Proceso BFS:**

- Paso 1: Visitar A y añadir B, C y D a la cola. Q: [B, C, D]
- Paso 2: Desencolar B y añadir E y F a la cola. Q: [C, D, E, F]
- Paso 3: Desencolar C y añadir F a la cola (pero F ya está en la cola). Q: [D, E, F]
- Paso 4: Desencolar D y añadir G a la cola. Q: [E, F, G]
- Paso 5: Desencolar E y añadir H a la cola. Q: [F, G, H]
- Paso 6: Desencolar F y añadir H a la cola (pero H ya está en la cola). Q: [G, H]
- Paso 7: Desencolar G y añadir I a la cola. Aquí encontramos I. Q: [H, I]



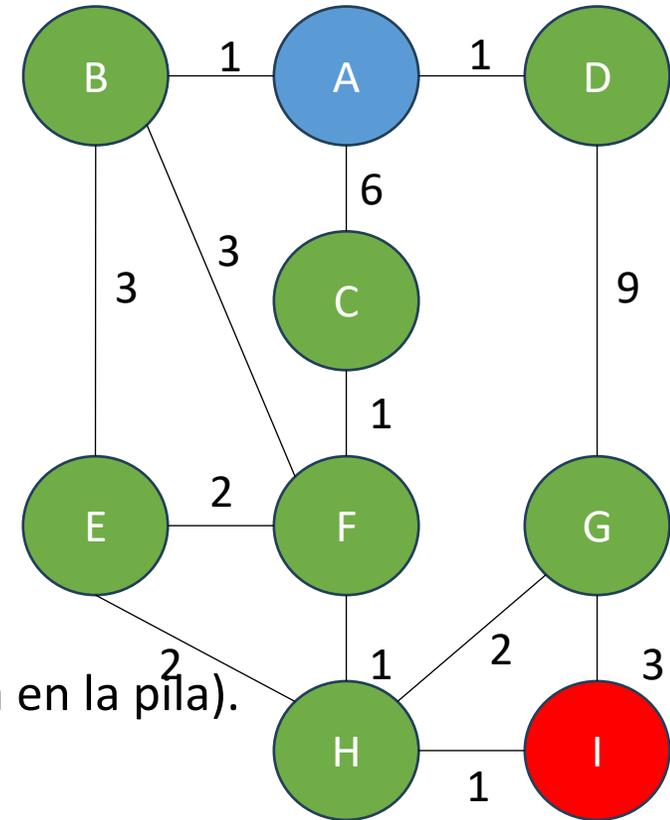
Ejercicio 2

- DFS:

- Comienza en el nodo A.
- Explora lo más profundamente posible antes de retroceder.
- Continúa hasta encontrar el nodo I.

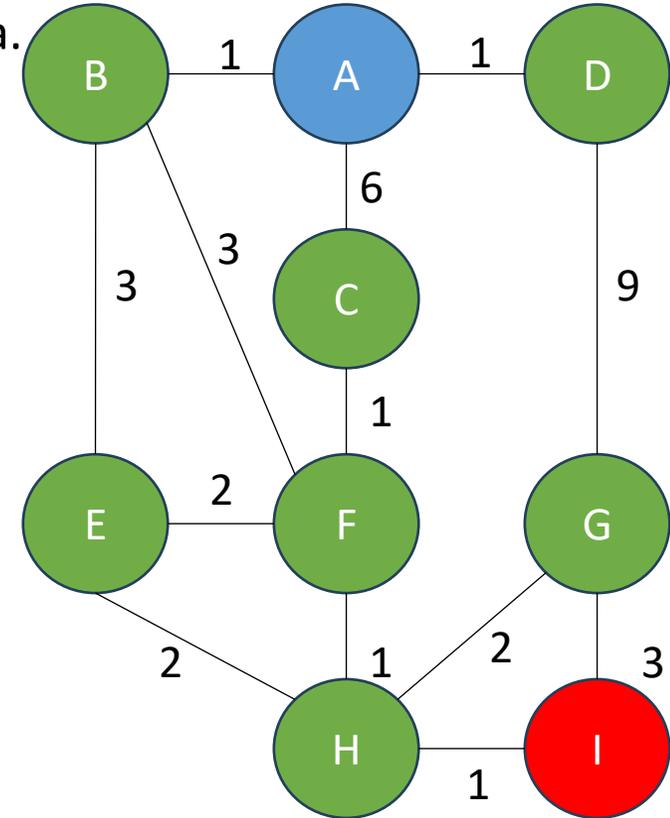
- Proceso DFS:

- Paso 1: Comenzamos en A. Añadimos D, C, y B a la pila. S: [D, C, B]
- Paso 2: Sacamos B de la pila y exploramos sus sucesores F y E. S: [D, C, F, E]
- Paso 3: Sacamos E de la pila y exploramos sus sucesores H y F (pero F ya está en la pila). S: [D, C, F, H]
- Paso 4: Sacamos H de la pila y exploramos su sucesor I. S: [D, C, F, I]
- Paso 5: Sacamos I de la pila. Hemos encontrado el objetivo. S: [D, C, F]

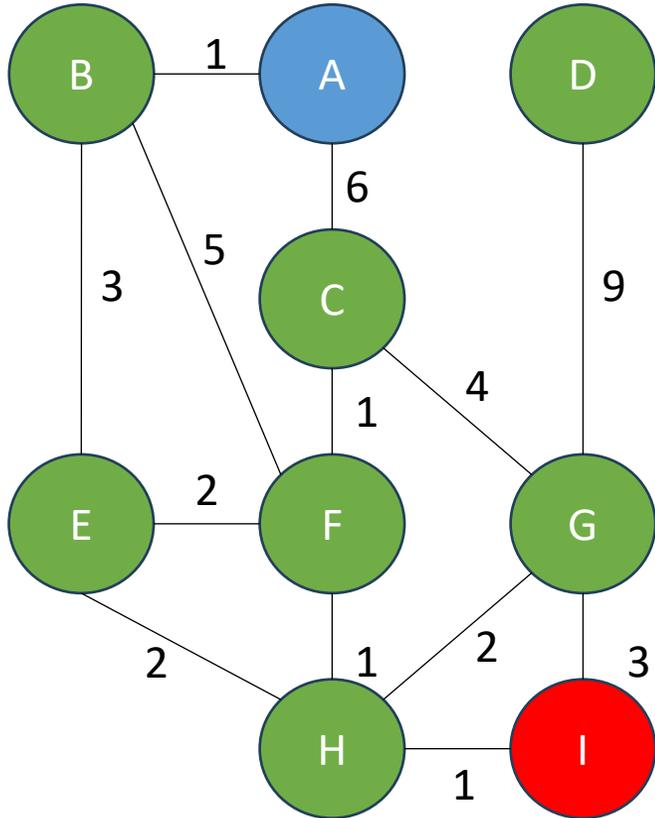


Ejercicio 2

- Paso 1: Comenzamos en A, añadimos B (coste 1), D (coste 1), C (coste 6) a la cola.
PQ: [(B, 1), (D, 1), (C, 6)]
- Paso 2: Desencolamos B, añadimos E (coste 3), F (coste 4) a la cola.
PQ: [(D, 1), (E, 3), (F, 4), (C, 6)]
- Paso 3: Desencolamos D, añadimos G (coste 10) a la cola.
PQ: [(E, 3), (F, 4), (C, 6), (G, 10)]
- Paso 4: Desencolamos E, añadimos H (coste 5), no se añade F ya que está.
Se actualiza C al haber encontrado un camino mejor.
PQ: [(F, 4), (H, 5), (C, 5), (G, 10)]
- Paso 5: Desencolamos F, no se añade H ya que está.
PQ: [(H, 5), (C, 5), (G, 10)]
- Paso 6: Desencolamos H, añadimos I (coste 6) a la cola. Aquí encontramos I.
PQ: [(C, 5), (I, 6), (G, 10)]
- Paso 7: Desencolamos C, no se añaden nodos.
PQ: [(I, 6), (G, 10)]
- Paso 8: Desencolamos I, no se añaden nodos. Hemos llegado al destino.
PQ: [(I, 6), (G, 10)]
- Resultado UCS: A -> B -> F -> H -> I (coste total: 6)



Ejercicio 3



- Sobre el grafo anterior:
 - Siendo A el vértice de inicio.
 - Siendo G el vértice objetivo.
- Se pide aplicar:
 - BFS
 - DFS
 - UCS

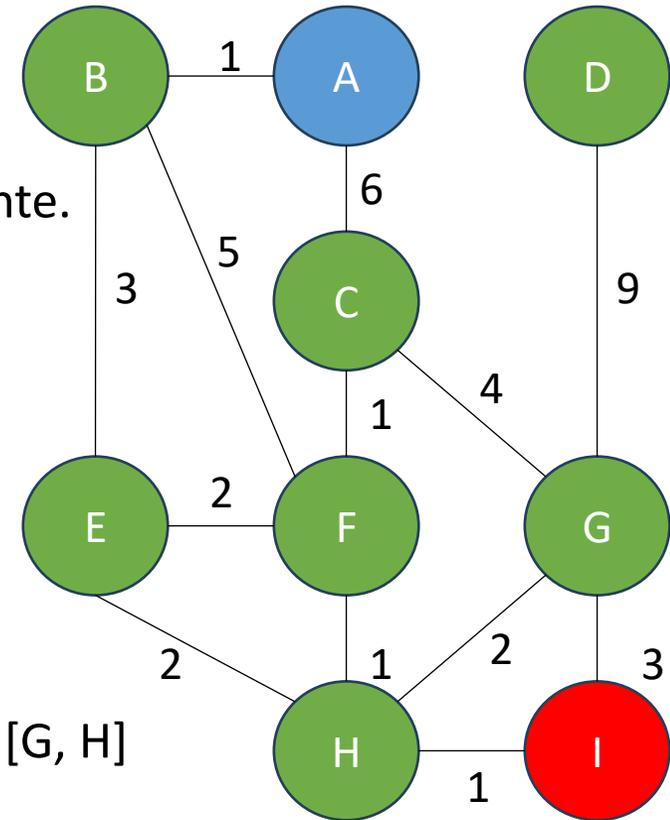
Ejercicio 3

- **BFS:**

- Comienza en el nodo A.
- Explora todos los nodos vecinos antes de avanzar a los nodos de nivel siguiente.
- Continúa hasta encontrar el nodo I.

- **Proceso BFS:**

- Paso 1: Comenzamos en A, añadimos B, C a la cola. Q: [B, C]
- Paso 2: Desencolamos B, añadimos E, F a la cola. Q: [C, E, F]
- Paso 3: Desencolamos C, añadimos G a la cola. Q: [E, F, G]
- Paso 4: Desencolamos E, añadimos H a la cola. Q: [F, G, H]
- Paso 5: Desencolamos F, no se añaden nodos porque H ya está en la cola. Q: [G, H]
- Paso 6: Desencolamos G, añadimos D e I a la cola. Q: [H, D, I]
- Paso 7: Desencolamos H, no se añaden nodos a la cola: Q: [D, I]
- Paso 8: Desencolamos D, no se añaden nodos a la cola: Q: [I]
- Paso 9: Desencolamos I, no se añaden nodos a la cola: Aquí encontramos I. Q: []



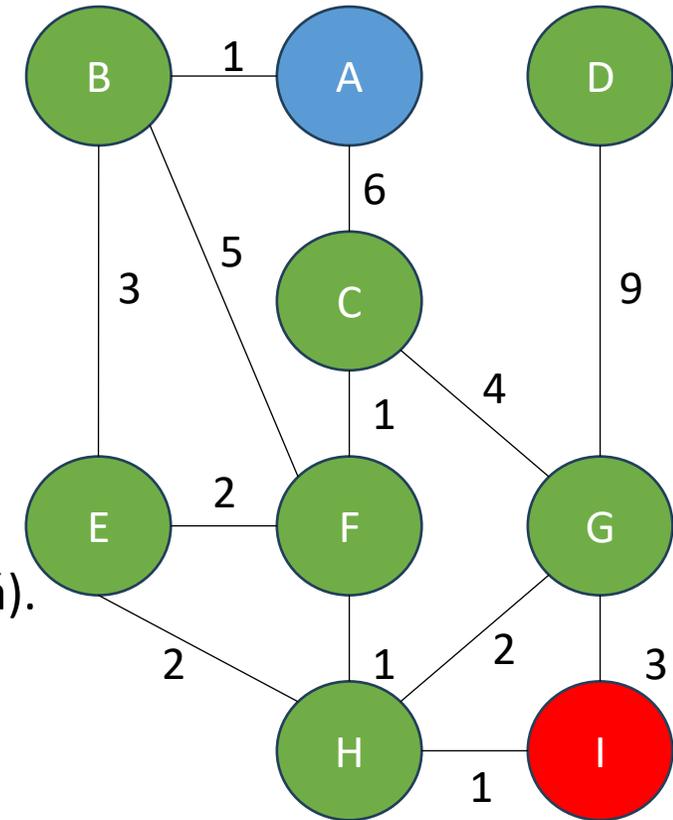
Ejercicio 3

- DFS:

- Comienza en el nodo A.
- Explora lo más profundamente posible antes de retroceder.
- Continúa hasta encontrar el nodo I.

- Proceso DFS:

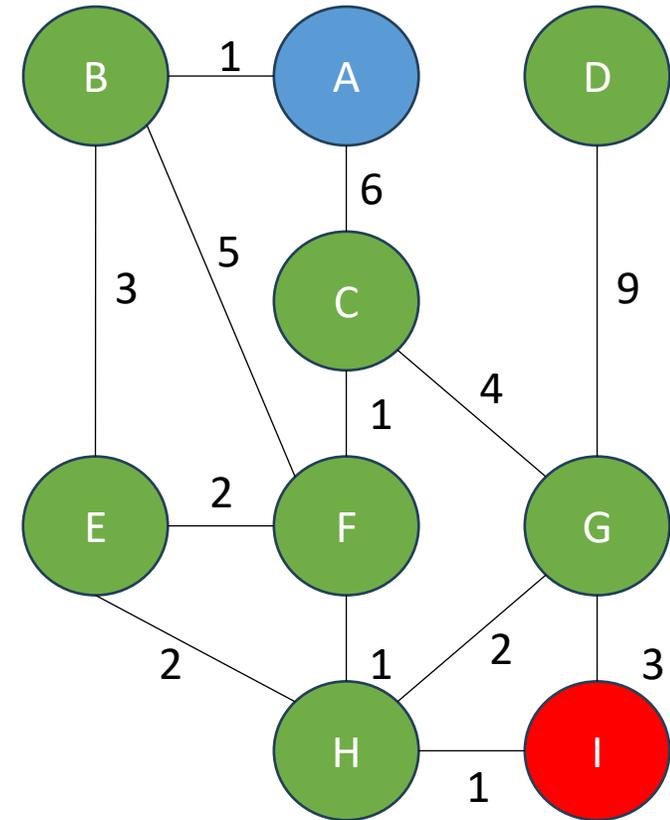
- Paso 1: Comenzamos en A. Añadimos C y B a la pila. S: [C, B]
- Paso 2: Sacamos B de la pila y exploramos sus sucesores F y E. S: [C, F, E]
- Paso 3: Sacamos E de la pila y exploramos sus sucesores H y F (pero F ya está). Pila: [C, F, H]
- Paso 4: Sacamos H de la pila y exploramos sus sucesores G e I. S: [C, F, I, G].
- Paso 5: Sacamos G de la pila, y no se añaden nodos. S: [C, F, I, D].
- Paso 6: Sacamos D de la pila, y no se añaden nodos. S: [C, F, I].
- Paso 7: Sacamos D de la pila, y no se añaden nodos. S: [C, F]. Se encuentra I.



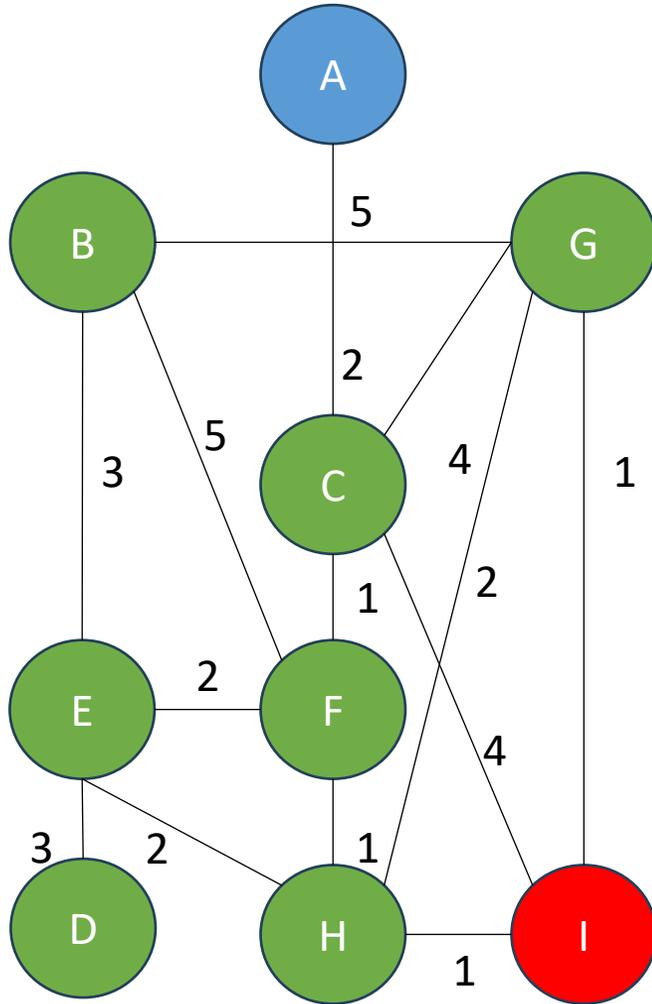
Ejercicio 3

• Proceso UCS:

- Paso 1: Comenzamos en A, añadimos B (coste 1), C (coste 6) a la cola.
PQ: [(B, 1), (C, 6)]
- Paso 2: Desencolamos B, añadimos E (coste 3), F (coste 6) a la cola.
PQ: [(E, 4), (C, 6), (F, 6)]
- Paso 3: Desencolamos E, añadimos H (coste 5), no se añade F ya que está en la cola.
PQ: [(C, 6), (F, 6), (H, 6)]
- Paso 4: Desencolamos C, se añade el nodo G.
PQ: [(F, 6), (H, 6), (G,10)]
- Paso 5: Desencolamos F, no se añaden nuevos nodos.
PQ: [(H, 6), (G,10)]
- Paso 6: Desencolamos H, se añade I, y actualiza G.
PQ: [(I, 7), (G,8)]
- Paso 7: Desencolamos G. Se ha llegado al destino.
PQ: [(G,8)]
- Camino Encontrado: A -> B -> E -> H -> I (coste Total: 7)



Ejercicio 4



- Sobre el grafo anterior:
 - Siendo A el vértice de inicio.
 - Siendo G el vértice objetivo.
- Se pide aplicar:
 - BFS
 - DFS
 - UCS

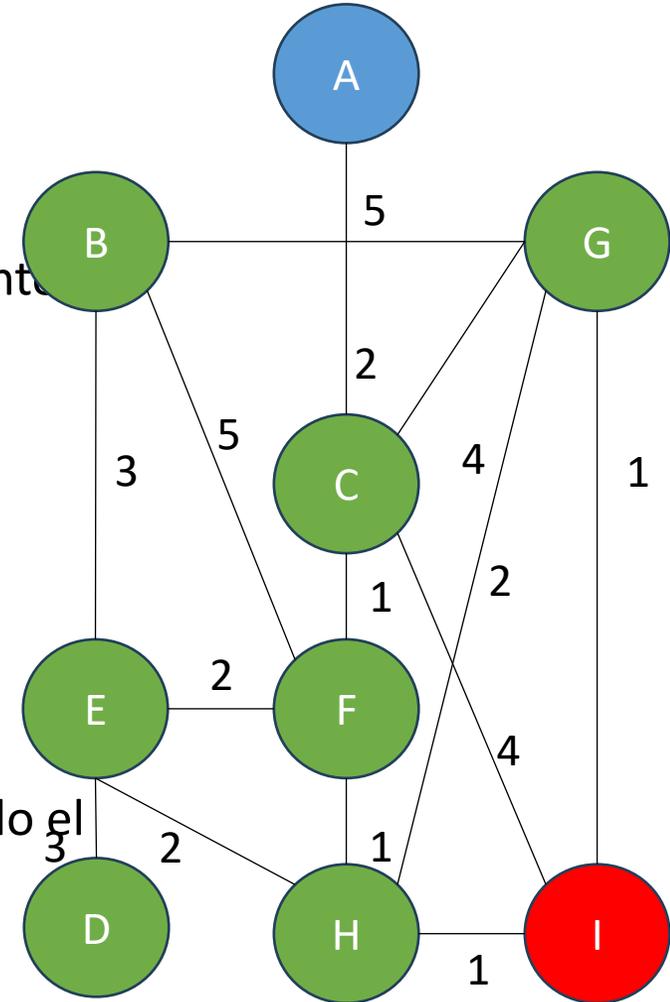
Ejercicio 4

- **BFS:**

- Comienza en el nodo A.
- Explora todos los nodos vecinos antes de avanzar a los nodos de nivel siguiente.
- Continúa hasta encontrar el nodo I.

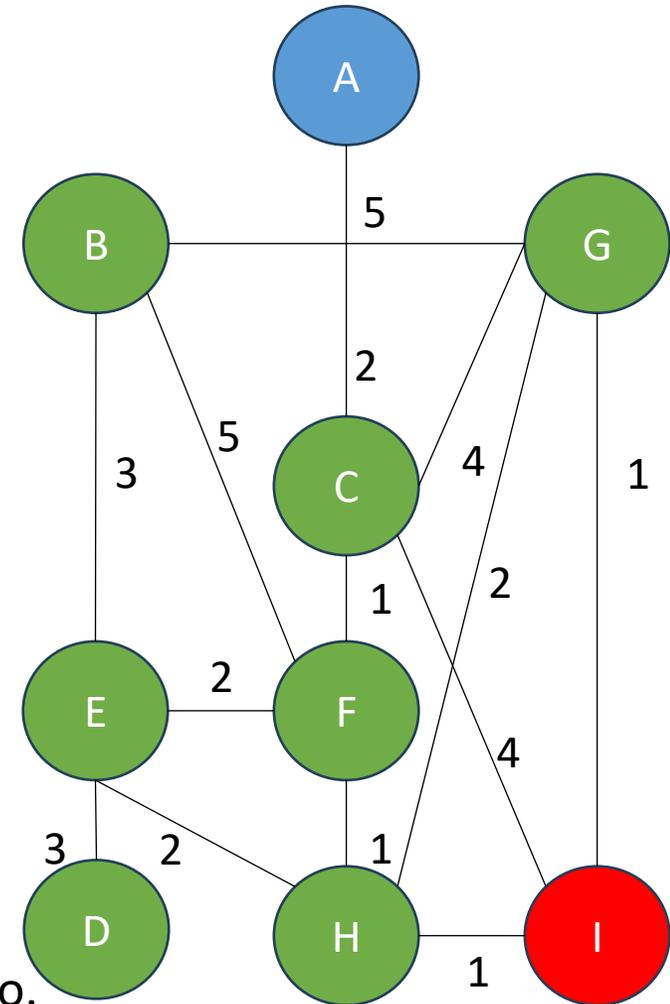
- **Proceso BFS:**

- Paso 1: Comenzamos en A. Añadimos C a la cola. Q: [C]
- Paso 2: Desencolamos C y exploramos sus sucesores F y G. Q: [F, G, I]
- Paso 3: Desencolamos F y exploramos sus sucesores. Q: [G, I, B, E, H]
- Paso 4: Desencolamos G y, no se añaden nodos. Q: [I, B, E, H]
- Paso 5: Desencolamos I y, no se añaden nodos. Q: [B, E, H]. Hemos encontrado el objetivo.



Ejercicio 4

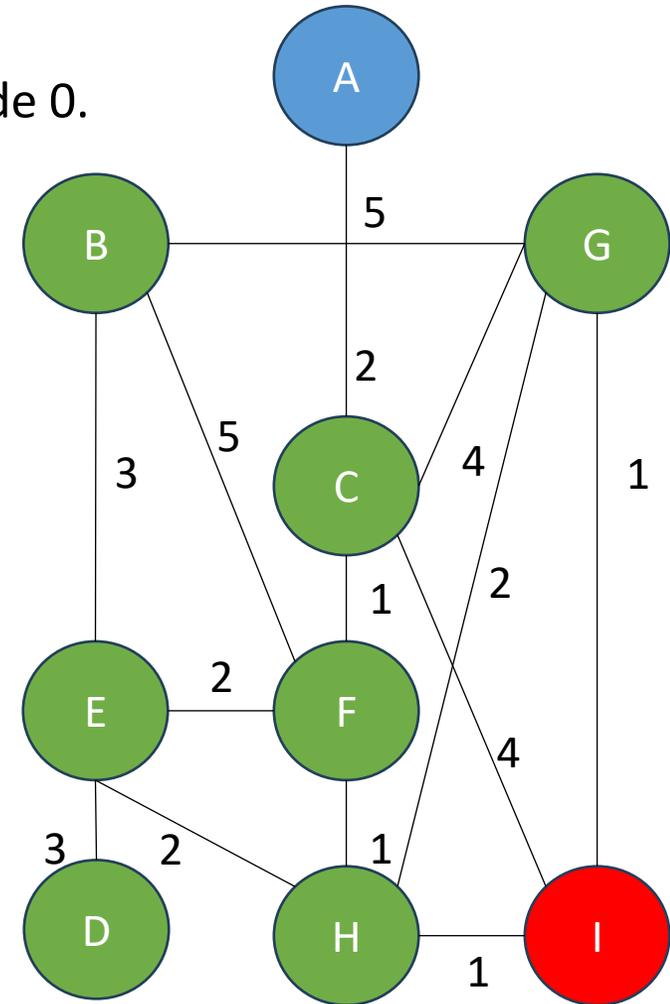
- DFS:
 - Comienza en el nodo A.
 - Explora lo más profundamente posible antes de retroceder.
 - Continúa hasta encontrar el nodo I.
- Proceso DFS:
 - Paso 1: Comenzamos en A. Añadimos C a la pila. S: [C]
 - Paso 2: Sacamos C de la pila y exploramos sus sucesores G, F e I. S: [I, G, F]
 - Paso 3: Sacamos F de la pila y exploramos su sucesor H. S: [I, G, H, E, B]
 - Paso 4: Sacamos B de la pila y no añadimos nuevos nodos. S: [I, G, H, E]
 - Paso 5: Sacamos E de la pila y, exploramos su sucesor D. S: [I, G, H, D]
 - Paso 6: Sacamos D de la pila y, no añadimos nuevos nodos. S: [I, G, H]
 - Paso 7: Sacamos H de la pila y, no añadimos nuevos nodos. S: [I, G]
 - Paso 8: Sacamos G de la pila y, no añadimos nuevos nodos. S: [I]
 - Paso 9: Sacamos I de la pila y, no añadimos nuevos nodos. Objetivo alcanzado.



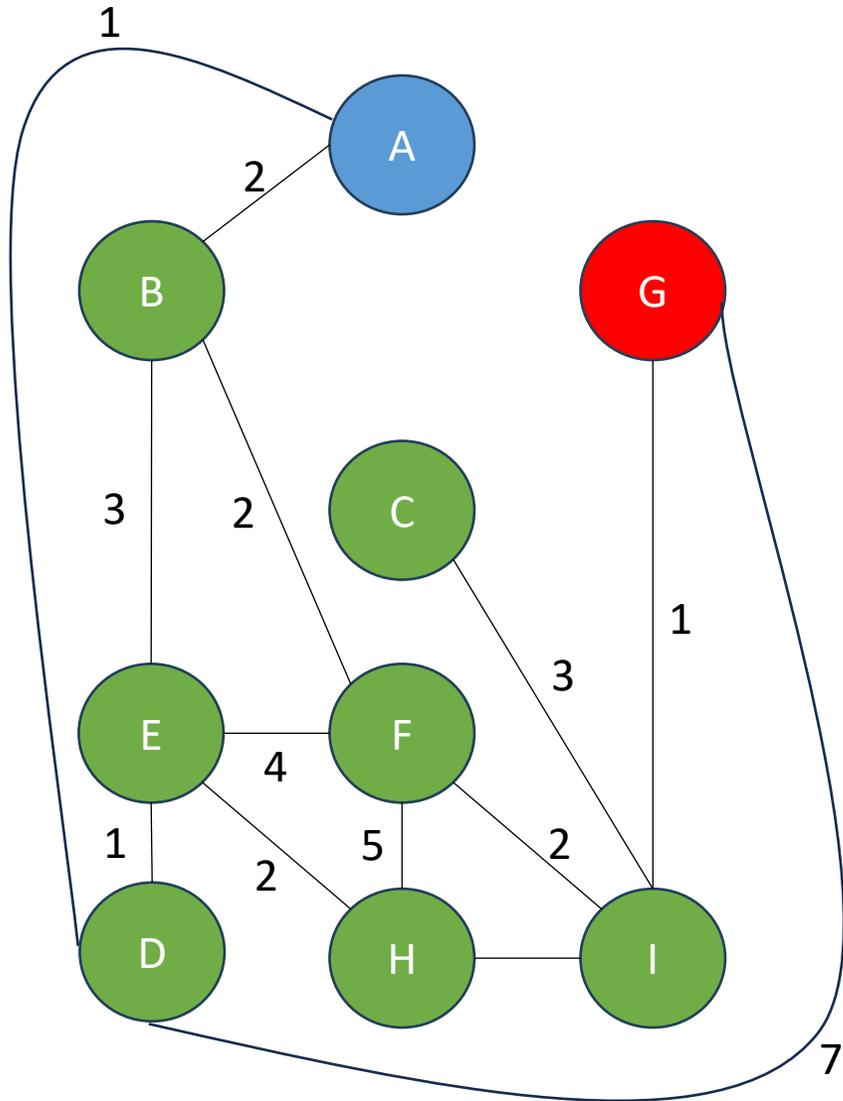
Ejercicio 4

• Proceso UCS:

- Paso 1: Inicialmente, solo el nodo A está en la cola con un coste acumulado de 0.
Q: [(A, 0)]
- Paso 2: Expandimos A. Los sucesores C se añaden a la cola con sus respectivos costes acumulados.
Q: [(C, 2)]
- Paso 3: Expandimos C. Añadimos sus sucesores F con un coste acumulado de 3 y G con un coste acumulado de 6.
Q: [(F, 3), (G, 6)]
- Paso 4: Expandimos F. Añadimos su sucesor H con un coste acumulado de 4.
Q: [(H, 4), (E, 5), (G, 6), (B, 8)]
- Paso 5: Expandimos H. Añadimos su sucesor I con un coste acumulado de 5.
Q: [(E, 5), (I, 5), (G, 6), (B, 8)]
- Paso 6: Expandimos E. Añadimos D.
Q: [(I, 5), (G, 6), (B, 8), (D, 8)]
- Paso 7: Expandimos I. Objetivo alcanzado.
Q: [(G, 6), (B, 8), (D, 8)]
- Camino encontrado: A -> C -> F -> H -> I (coste total: 5)



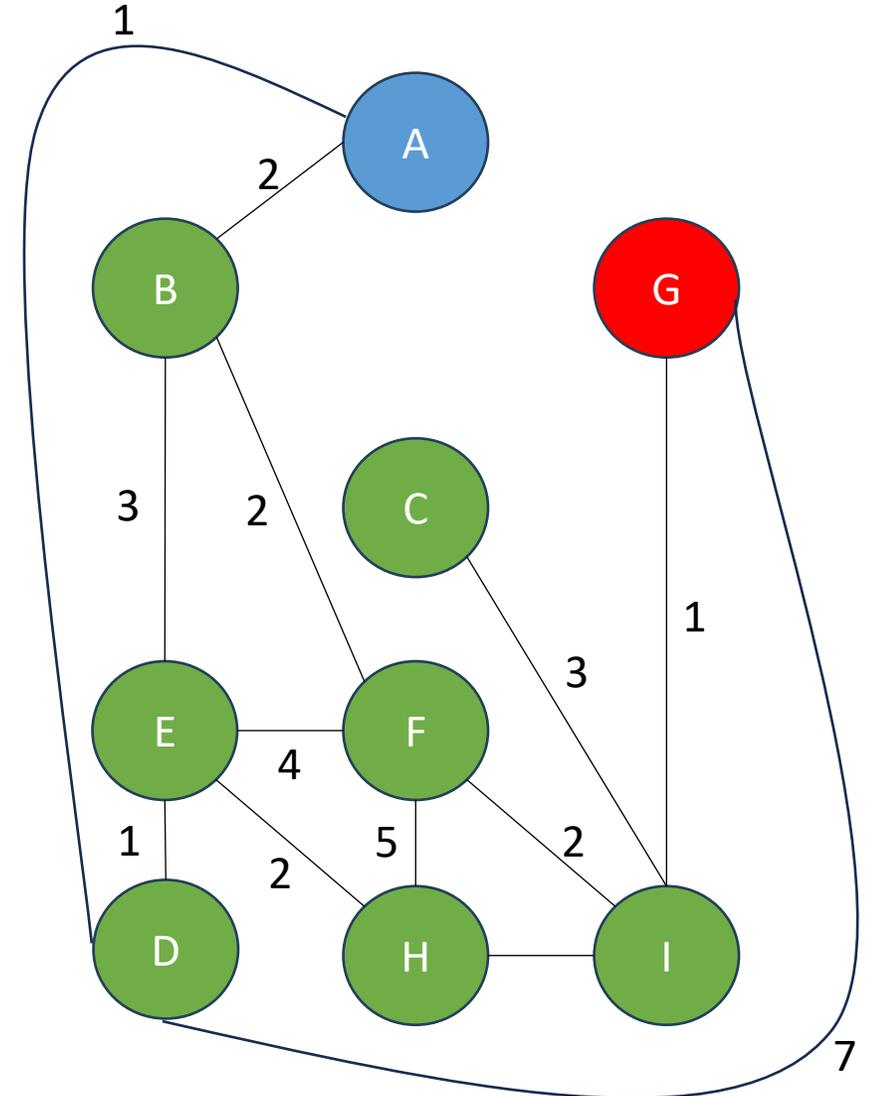
Ejercicio 5



- Sobre el grafo anterior:
 - Siendo A el vértice de inicio.
 - Siendo G el vértice objetivo.
- Se pide aplicar:
 - BFS
 - DFS
 - UCS

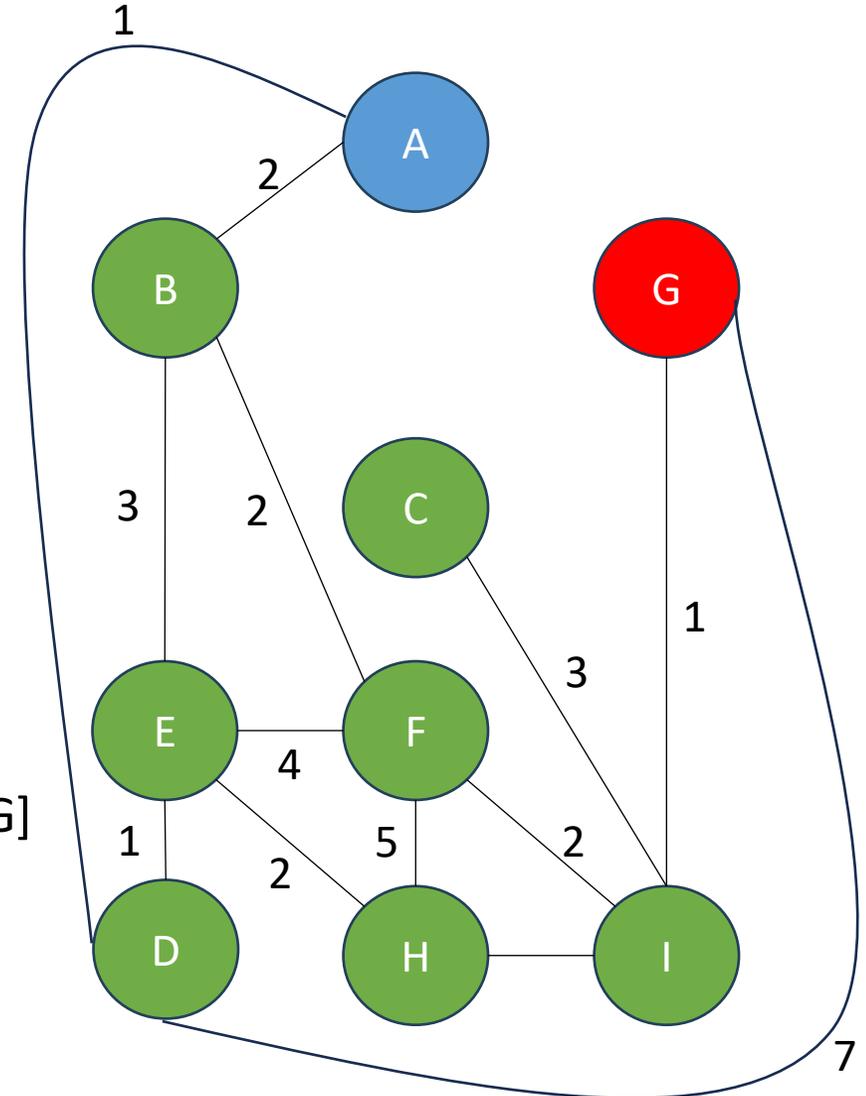
Ejercicio 5

- **BFS:**
 - Comienza en el nodo A.
 - Explora todos los nodos vecinos antes de avanzar a los nodos de nivel siguiente.
 - Continúa hasta encontrar el nodo I.
- **Proceso BFS:**
 - Paso 1: Comenzamos en A, añadimos B, D a la cola. Q: [B, D]
 - Paso 2: Desencolamos B, añadimos E, F a la cola. Q: [D, E, F]
 - Paso 3: Desencolamos D, añadimos G. Q: [E, F, G]
 - Paso 4: Desencolamos E, añadimos H a la cola. Q: [F, G, H]
 - Paso 5: Desencolamos F, añadimos I a la cola. Q: [G, H, I]
 - Paso 6: Desencolamos G, no se añaden nuevos nodos. Q: [H, I]



Ejercicio 5

- DFS:
 - Comienza en el nodo A.
 - Explora lo más profundamente posible antes de retroceder.
 - Continúa hasta encontrar el nodo I.
- Proceso DFS:
 - Paso 1: Comenzamos en A, añadimos D, B a la pila. S: [D, B]
 - Paso 2: Sacamos B de la pila, añadimos F, E a la pila. S: [D, F, E]
 - Paso 3: Sacamos E de la pila, añadimos H a la pila. S: [D, F, H]
 - Paso 4: Sacamos H de la pila, añadimos I a la pila. S: [D, F, I]
 - Paso 5: Sacamos I de la pila, añadimos G, C a la pila. S: [D, F, G, C]
 - Paso 6: Sacamos C de la pila, no se añaden nuevos nodos. S: [D, F, G]
 - Paso 7: Sacamos G de la pila, no se añaden nuevos nodos. S: [D, F].
Objetivo alcanzado.



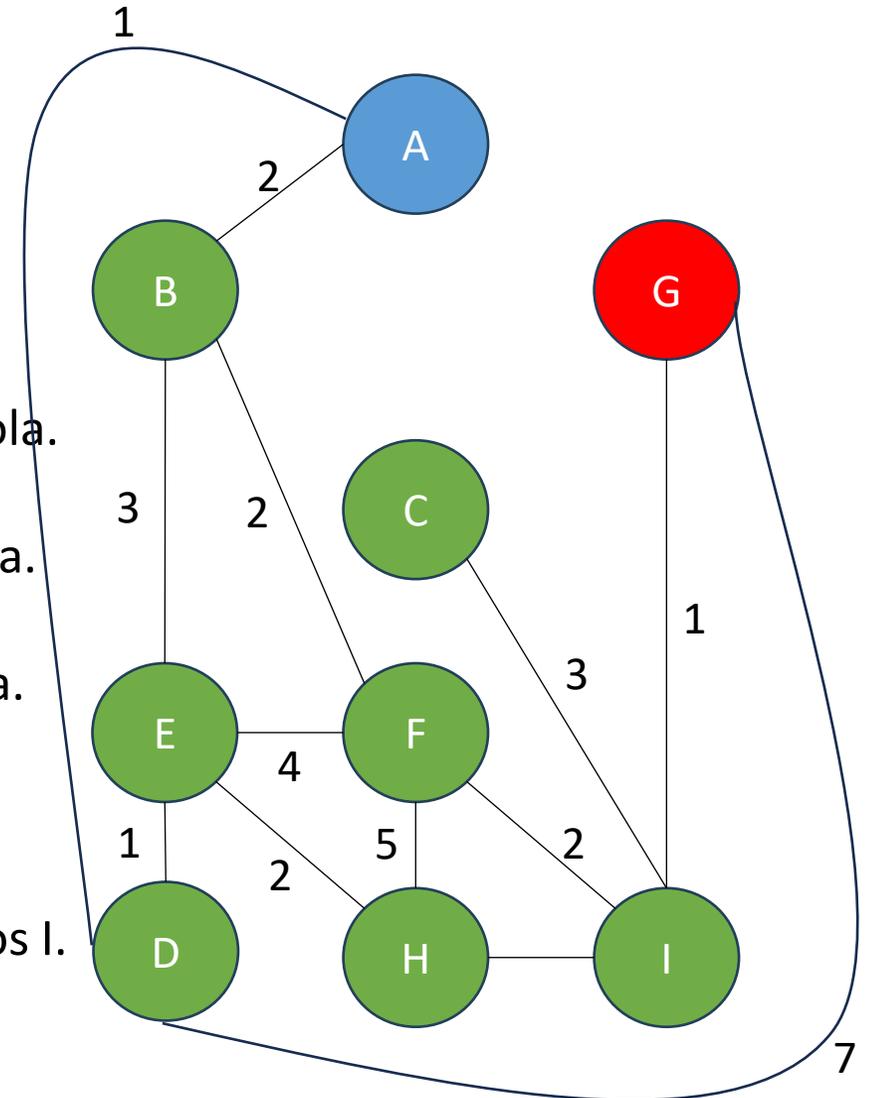
Ejercicio 5

- UCS:

- Comienza en el nodo A.
- Explora lo más profundamente posible antes de retroceder.
- Continúa hasta encontrar el nodo I.

- Proceso UCS:

- Paso 1: Comenzamos en A, añadimos D (coste 1), B (coste 2) a la cola.
PQ: [(D, 1), (B, 2)]
- Paso 2: Desencolamos D, añadimos G (coste 8) y E (coste 2) a la cola.
PQ: [(B, 2), (E, 2), (G, 8)]
- Paso 3: Desencolamos B, añadimos F (coste 4) y E (coste 5) a la cola.
PQ: [(E, 2), (F, 4), (E, 5), (G, 8)]
- Paso 4: Desencolamos E, añadimos H (coste 4) a la cola.
PQ: [(F, 4), (H, 4), (E, 5), (G, 8)]
- Paso 5: Desencolamos F, añadimos I (coste 6) a la cola. Encontramos I.
PQ: [(H, 4), (E, 5), (G, 8)]
- Resultado UCS : A -> D -> E -> H -> I (coste Total: 4)



DESARROLLO DE JUEGOS CON INTELIGENCIA ARTIFICIAL

Búsqueda en el espacio de estados y agentes

Tema 2. Búsqueda no informada

©2023 Autores
David María Arribas,
Sergio Hernández García,
Nicolás H. Rodríguez Uribe,

Algunos derechos reservados
Este documento se distribuye bajo la licencia
"Atribución-CompartirIgual 4.0 Internacional" de Creative Commons, disponible en
<https://creativecommons.org/licenses/by-sa/4.0/deed.es>



Nicolás Rodríguez
nicolas.rodriguez@urjc.es

David María
david.maria@urjc.es

Sergio Hernández
sergio.Hernandez@urjc.es



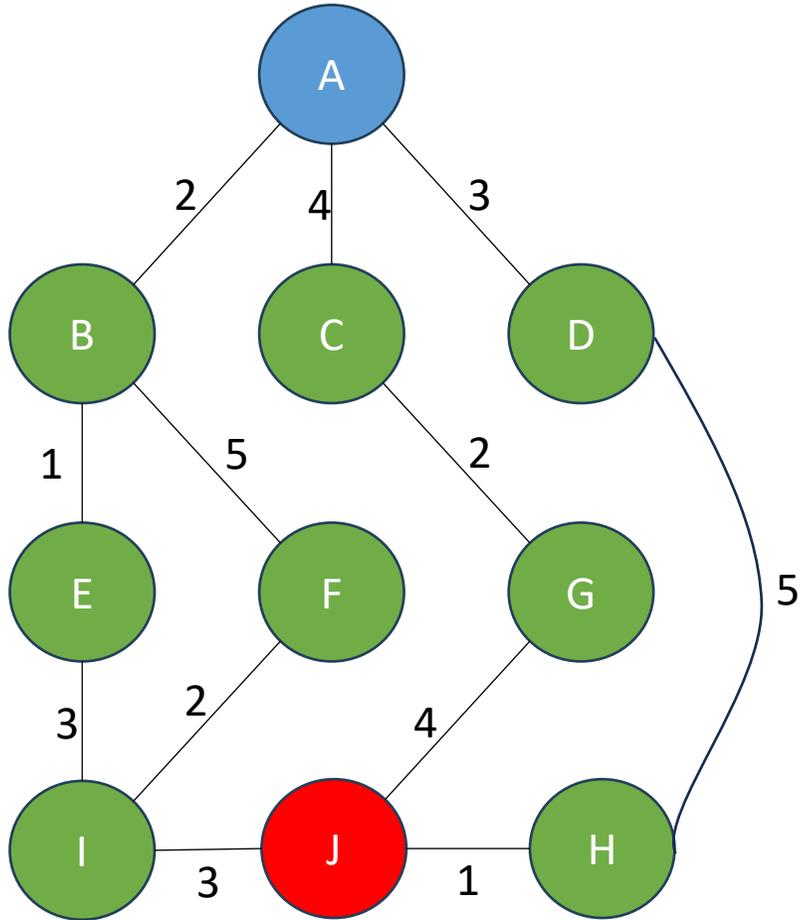
DESARROLLO DE JUEGOS CON INTELIGENCIA ARTIFICIAL

Búsqueda en el espacio de estados y agentes

Tema 3. Búsqueda con heurísticas débiles



Ejercicio 1

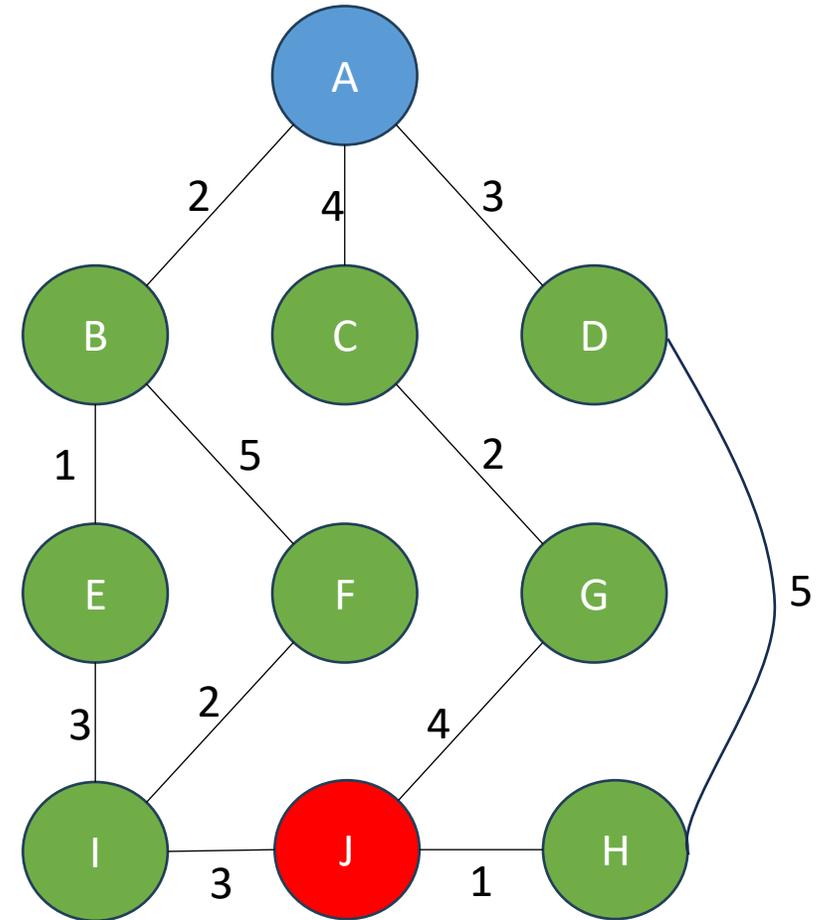


- Sobre el grafo anterior:
 - Siendo A el vértice de inicio.
 - Siendo J el vértice objetivo.
- Se pide aplicar:
 - Búsqueda por subobjetivos (D, H, J)
 - Búsqueda por ascenso de colinas
 - Búsqueda por horizonte (2)

Ejercicio 1

- Hill Climbing:

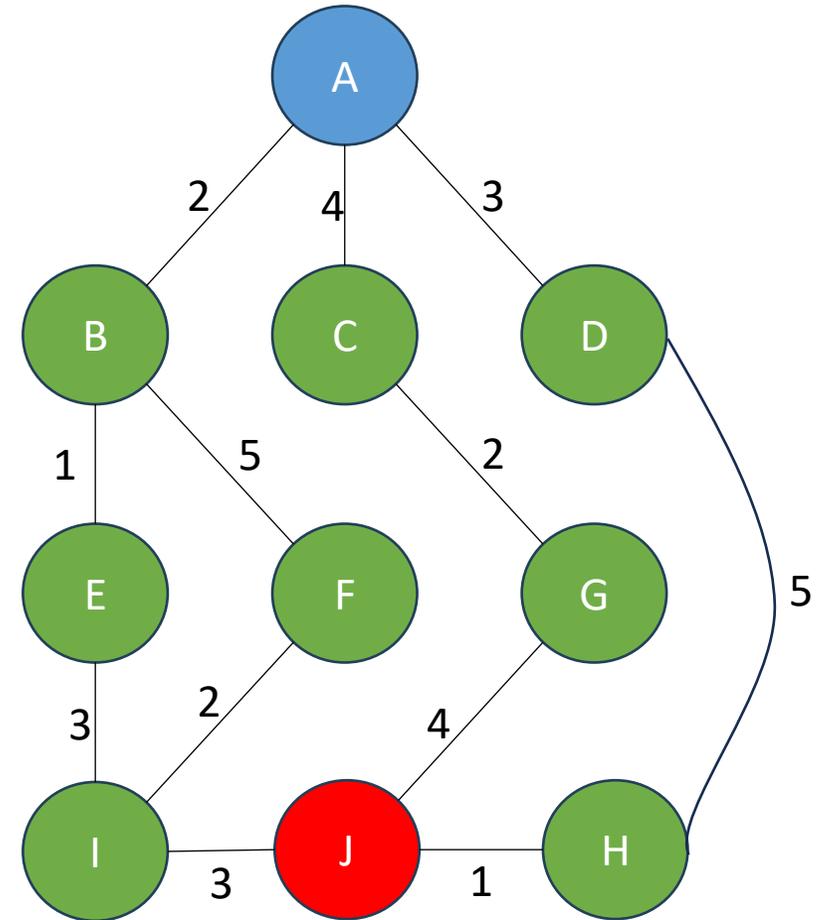
- Inicio en *A*: El algoritmo evalúa los vecinos de *A* para determinar cuál tiene la mejor heurística en términos de proximidad a *J*.
 - Vecinos desde *A*: *B* (peso 2), *C* (peso 4), y *D* (peso 3).
 - Se selecciona *B* porque tiene el menor peso (2).
- Desde *B*: El algoritmo evalúa los vecinos de *B* para seguir minimizando la distancia al objetivo.
 - Vecinos de *B*: *A* (visitado), *E* (peso 1), y *F* (peso 5).
 - Se selecciona *E* con un coste de 1.
- Desde *E*: El algoritmo evalúa los vecinos de *E*.
 - Vecinos de *E*: *B* (visitado) y *I* (peso 3).
 - Se selecciona *I* con un coste de 3.
- Desde *I*: El algoritmo evalúa sus vecinos..
 - Vecinos de *I*: *E* (visitado), *F* (peso 2), y *J* (peso 3).
 - Se selecciona *F* con un coste de 2.
- Se mueve a *F* y queda atrapado.



Ejercicio 1

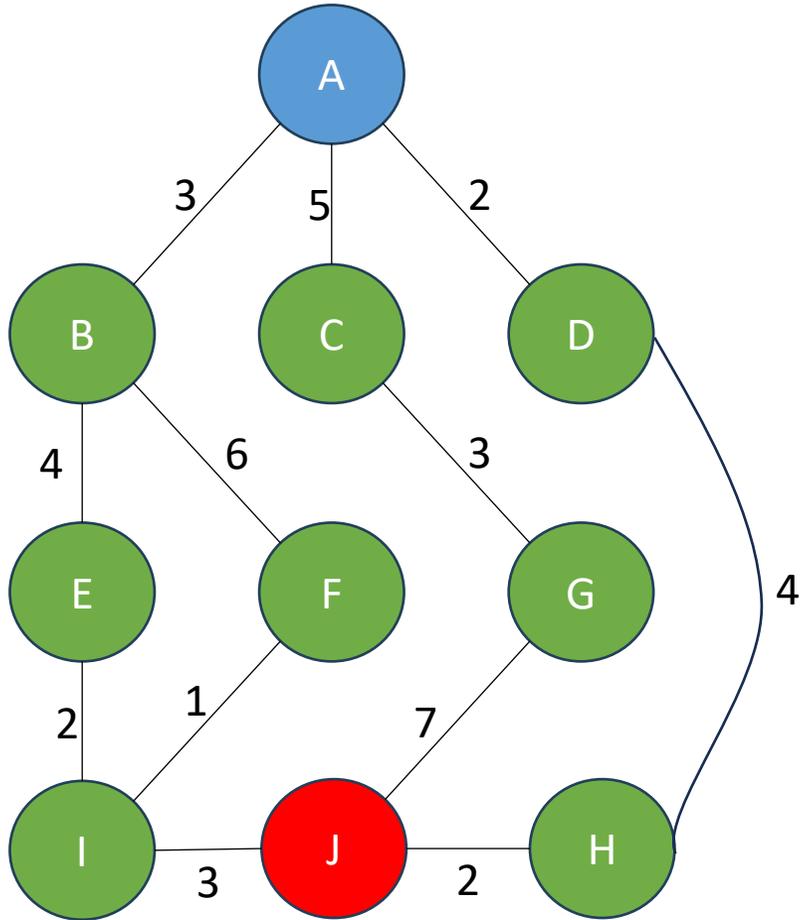
- Búsqueda por horizonte:

- Inicio en A : El algoritmo evalúa las posibles secuencias de movimientos dentro del horizonte (2 niveles).
 - Vecinos desde A : B, C, D .
 - Se simulan secuencias:
 - $A \rightarrow B \rightarrow E$ (coste: 3)
 - $A \rightarrow B \rightarrow F$ (coste: 7)
 - $A \rightarrow C \rightarrow G$ (coste: 6)
 - $A \rightarrow D \rightarrow H$ (coste: 8)
 - La secuencia $A \rightarrow B \rightarrow E$ parece la más prometedora.
- Desde B : Después de seleccionar B , el algoritmo evalúa los caminos desde B .
 - Vecinos de B : A (visitado), E (peso 3) y F (peso 5).
 - Se simulan secuencias:
 - $B \rightarrow E \rightarrow I$ (coste: 4)
 - $B \rightarrow F \rightarrow I$ (coste: 7)
 - La secuencia $B \rightarrow E \rightarrow I$ parece la más prometedora.
- Desde E : Finalmente, el algoritmo evalúa los vecinos de E .
 - Vecinos de E : B (visitado) e I (peso 3).
 - La secuencia $E \rightarrow I \rightarrow J$ es la única posible



- Camino final: $A \rightarrow B \rightarrow E \rightarrow I \rightarrow J$
- Coste total: $2+1+3+3=9$

Ejercicio 2

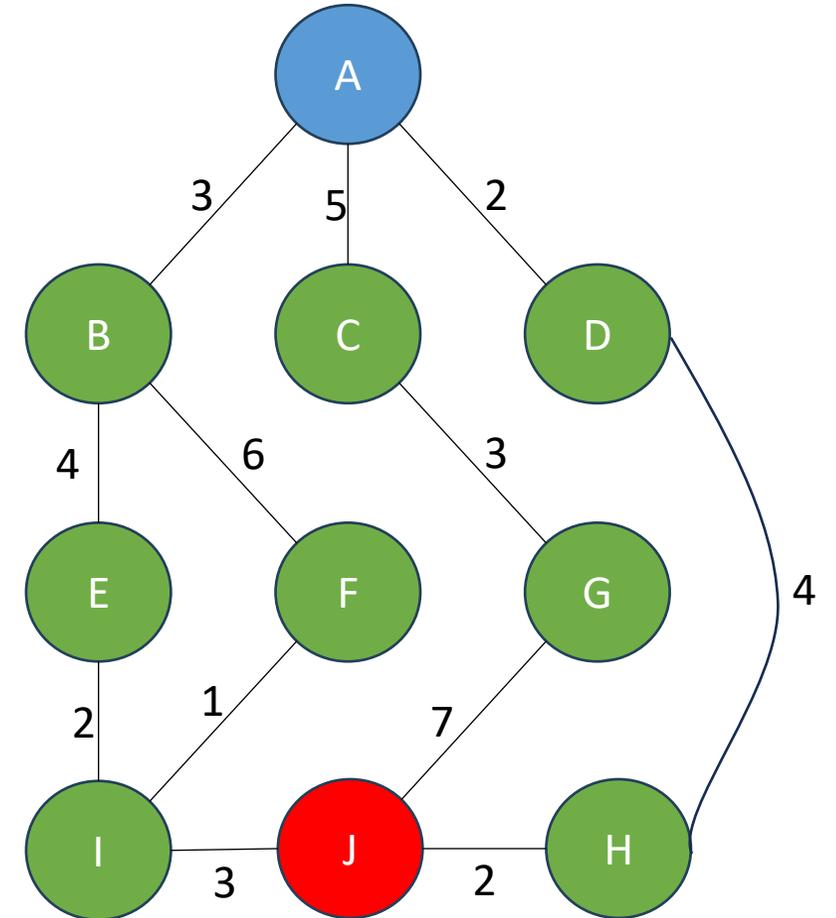


- Sobre el grafo anterior:
 - Siendo A el vértice de inicio.
 - Siendo J el vértice objetivo.
- Se pide aplicar:
 - Búsqueda por subobjetivos (D, H, J)
 - Búsqueda por ascenso de colinas
 - Búsqueda por horizonte (2)

Ejercicio 2

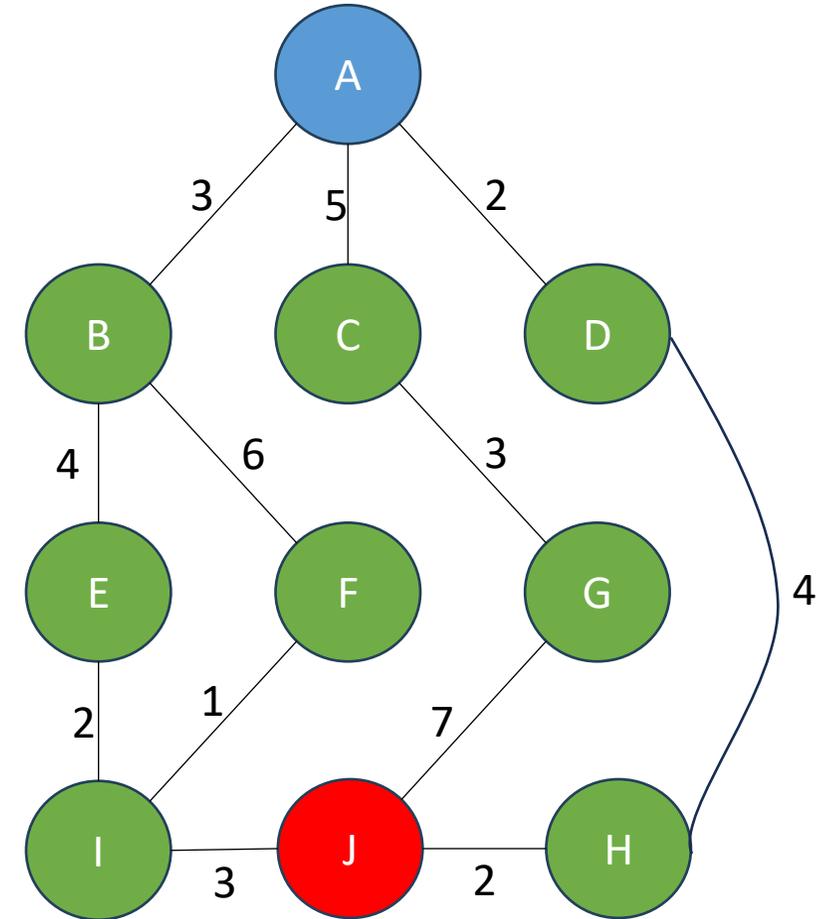
- Hill Climbing:

- Inicio en *A*:
 - Posibles vecinos desde *A*: *B* (peso 3), *C* (peso 5), y *D* (peso 2).
 - Se selecciona *D* por ser el de menor coste 2.
- Desde *D*:
 - Vecinos de *D*: *A* (visitado) y *H* (peso 4).
 - Se selecciona *H* con un coste de 4.
- Desde *H*:
 - Vecinos de *H*: *D* (visitado) y *J* (peso 2).
 - Se selecciona *J* con un coste de 1.
- Camino final: $A \rightarrow D \rightarrow H \rightarrow J$
- Coste total: $2+4+2=8$



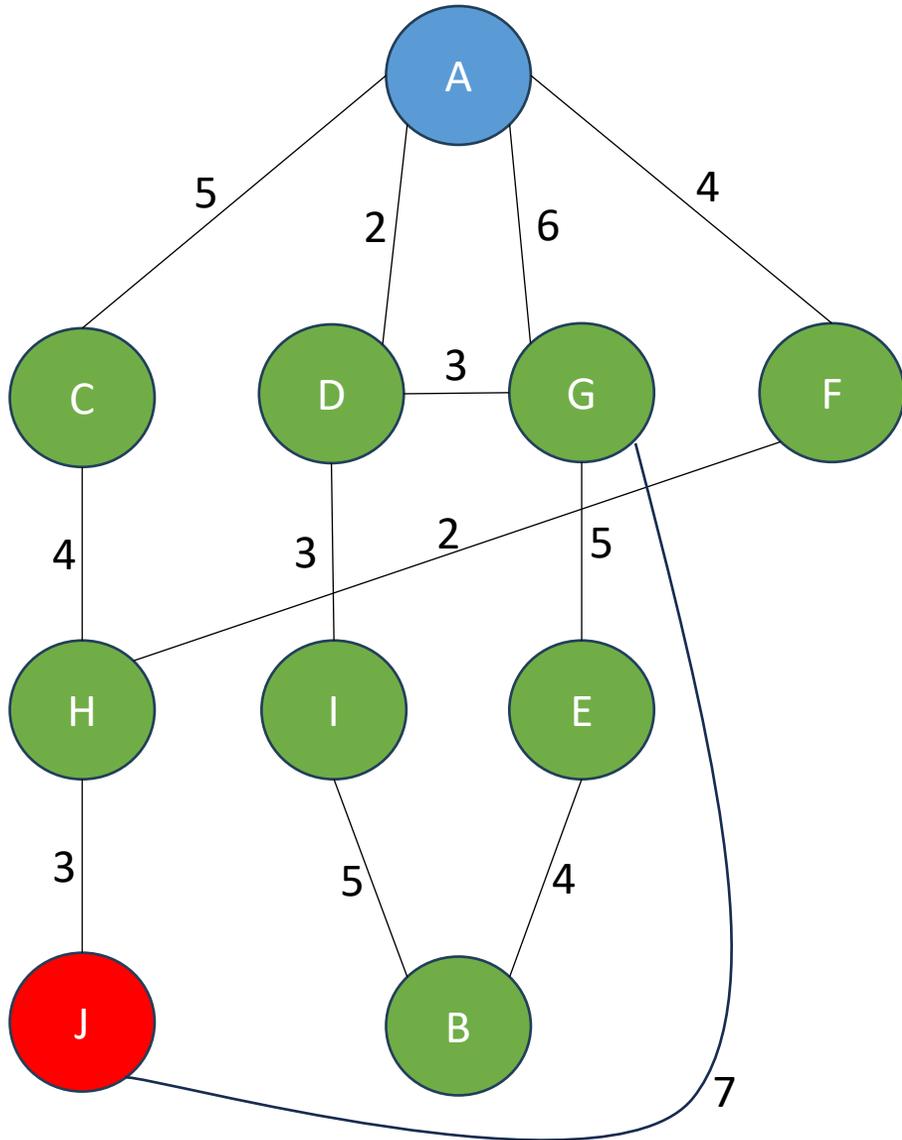
Ejercicio 2

- **Búsqueda por horizonte:**
 - Inicio en A : El algoritmo evalúa las posibles secuencias de movimientos dentro del horizonte (2 niveles).
 - Vecinos desde A : B, C, D .
 - Se simulan secuencias:
 - $A \rightarrow B \rightarrow E$ (coste: 7)
 - $A \rightarrow B \rightarrow F$ (coste: 9)
 - $A \rightarrow C \rightarrow G$ (coste: 8)
 - $A \rightarrow D \rightarrow H$ (coste: 6)
 - La secuencia $A \rightarrow D \rightarrow H$ parece la más prometedora.
 - Desde D : Después de seleccionar D , el algoritmo evalúa los caminos desde D .
 - Vecinos de D : A (visitado), y H (peso 4).
 - La secuencia $D \rightarrow H \rightarrow J$ es la única posible.
 - Desde H : Finalmente, el algoritmo evalúa los vecinos de H .
 - Vecinos de H : D (visitado) e J (peso 2).
 - La secuencia $H \rightarrow J$ es la única posible



- Camino final: $A \rightarrow D \rightarrow H \rightarrow J$
- Coste total: $2+4+2=8$

Ejercicio 3

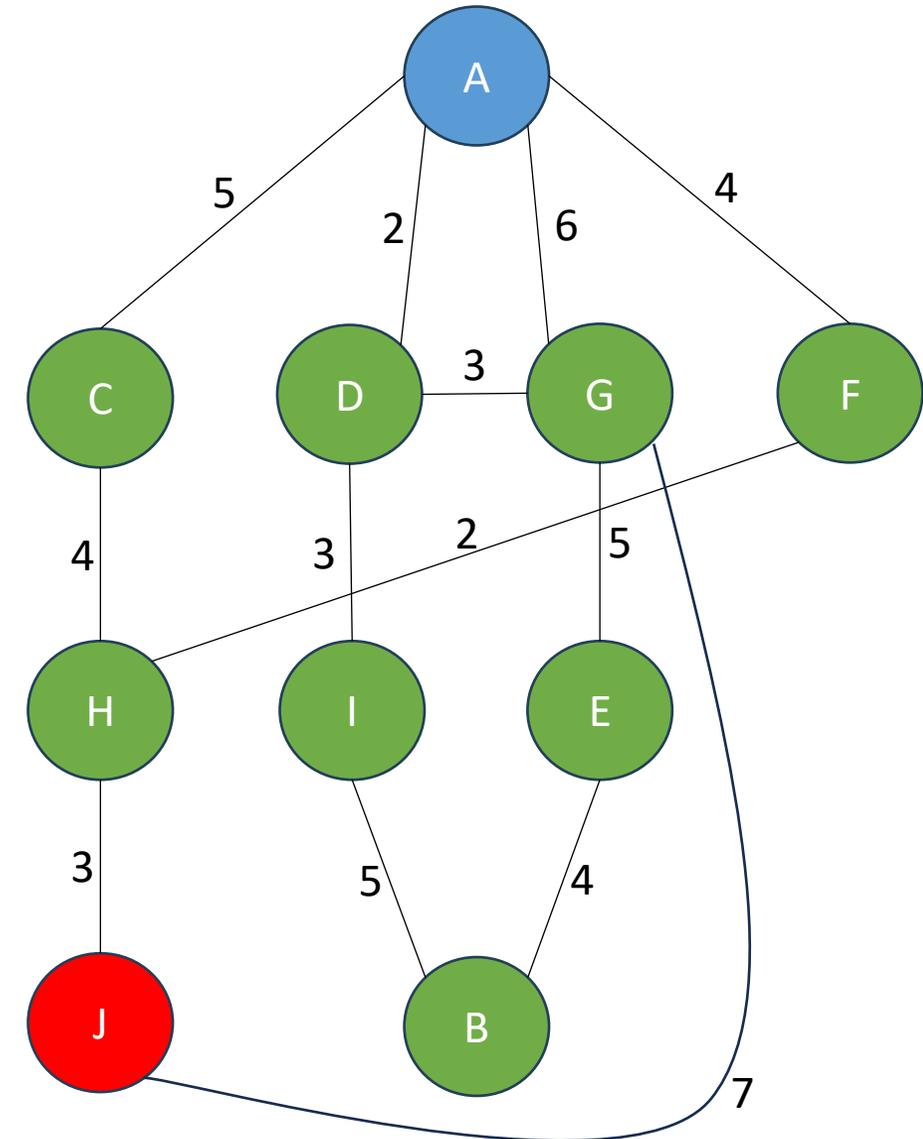


- Sobre el grafo anterior:
 - Siendo A el vértice de inicio.
 - Siendo J el vértice objetivo.
- Se pide aplicar:
 - Búsqueda por subobjetivos (D, G, J)
 - Búsqueda por ascenso de colinas
 - Búsqueda por horizonte (2)

Ejercicio 3

• Hill Climbing:

- Inicio en *A*:
 - Vecinos de *A*: *C* (peso 5), *D* (peso 2), *F* (peso 4) y *G* (peso 6).
 - Se selecciona *D* con un coste de 2.
- Desde *D*:
 - Vecinos de *D*: *A* (visitado), *G* (peso 3), *I* (peso 3).
 - Se selecciona *G* con un coste de 3.
- Desde *G*:
 - Vecinos de *G*: *A* (visitado), *D* (visitado), *E* (peso 5), *J* (peso 7).
 - Se selecciona *E* con un coste de 5.
- Desde *E*:
 - Vecinos de *E*: *G* (visitado), *B* (peso 4).
 - Se selecciona *E* con un coste de 5.
- Desde *I*:
 - Vecinos de *E*: *D* (visitado), *B* (visitado).
- Se queda atrapado en una meseta

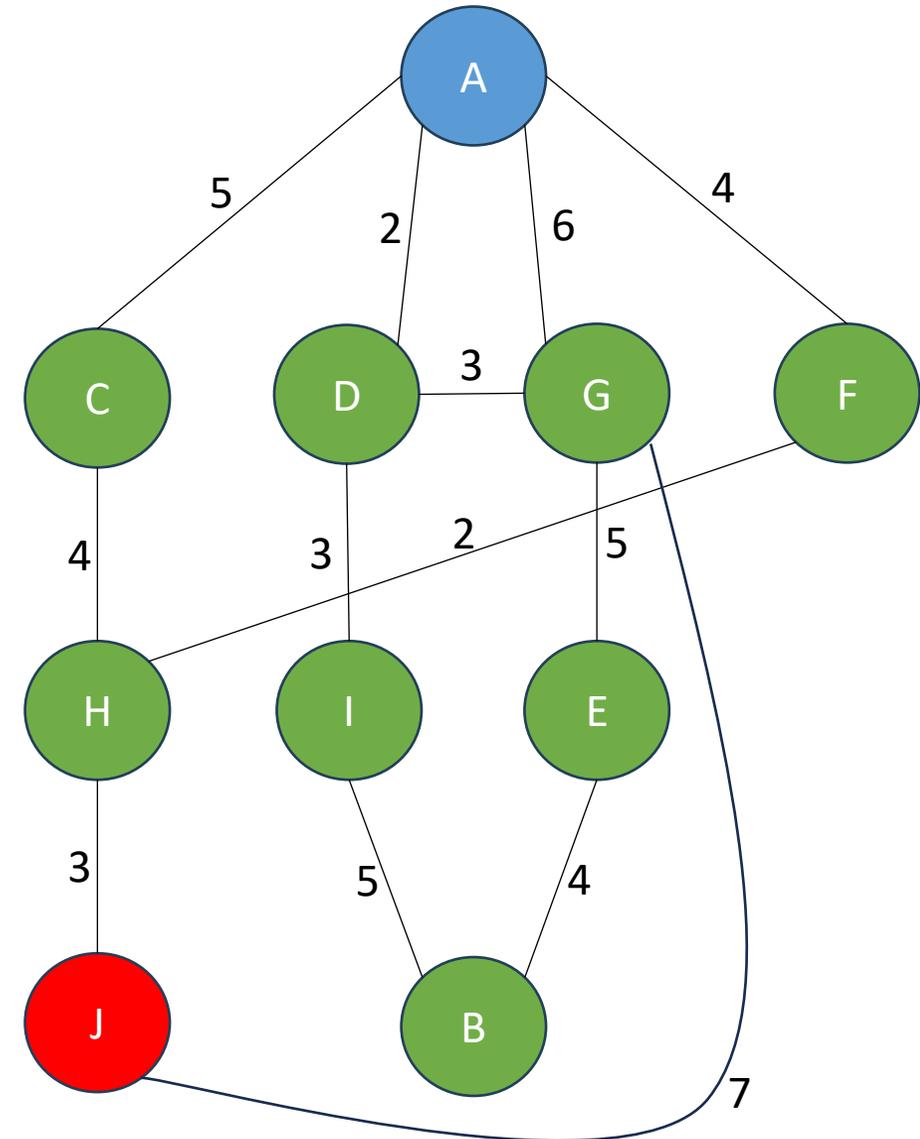


Ejercicio 3

- **Búsqueda por horizonte:**

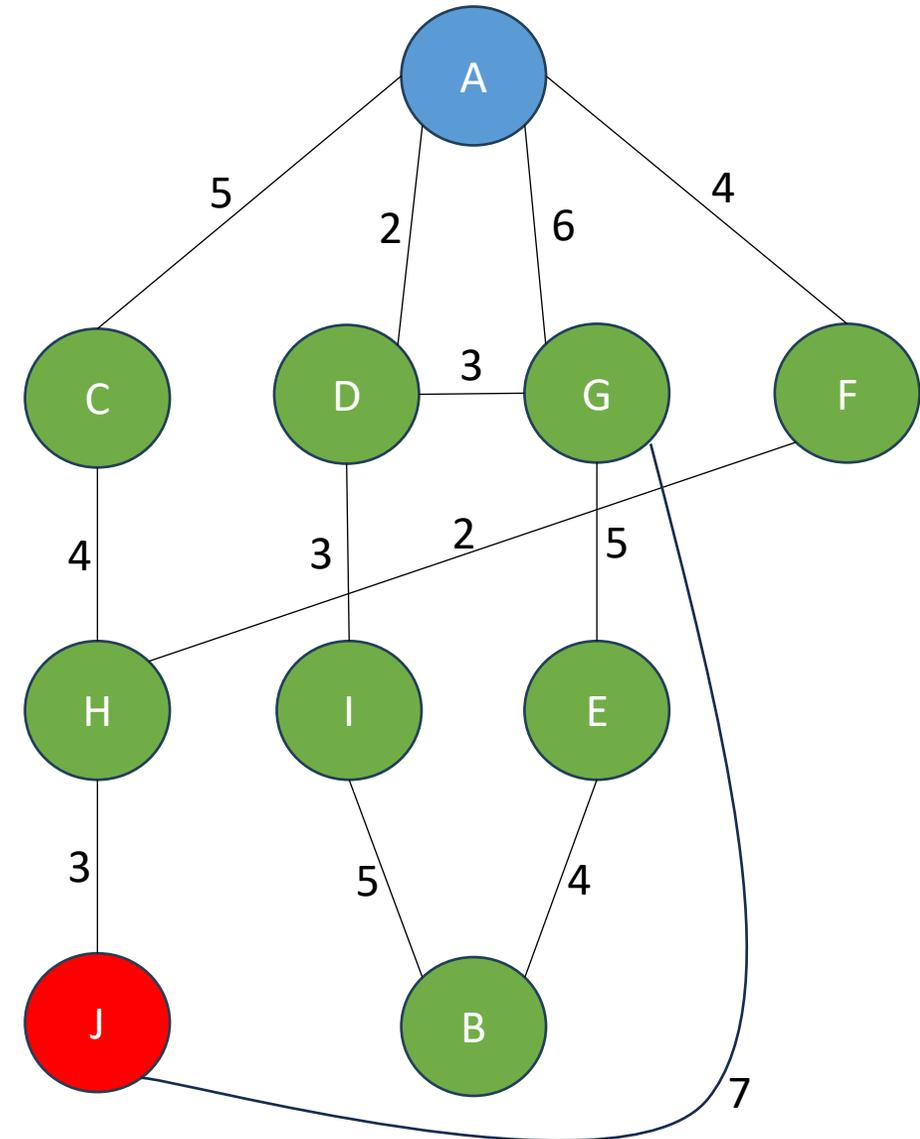
- Inicio en A : El algoritmo evalúa las posibles secuencias de movimientos dentro del horizonte (2 niveles).

- Vecinos desde A : C , D , F y G
- Se simulan secuencias:
 - $A \rightarrow C \rightarrow H$ (coste: 9)
 - $A \rightarrow D \rightarrow I$ (coste: 5) (primero en la cola)
 - $A \rightarrow D \rightarrow G$ (coste: 5)
 - $A \rightarrow G \rightarrow D$ (coste: 9)
 - $A \rightarrow G \rightarrow E$ (coste: 9)
 - $A \rightarrow C \rightarrow G$ (coste: 6)
 - $A \rightarrow F \rightarrow H$ (coste: 6)
- Las secuencias $A \rightarrow D \rightarrow I$ y $A \rightarrow D \rightarrow G$ parecen la más prometedoras.
- Se elige $A \rightarrow D \rightarrow I$ por estar antes en la cola.



Ejercicio 3

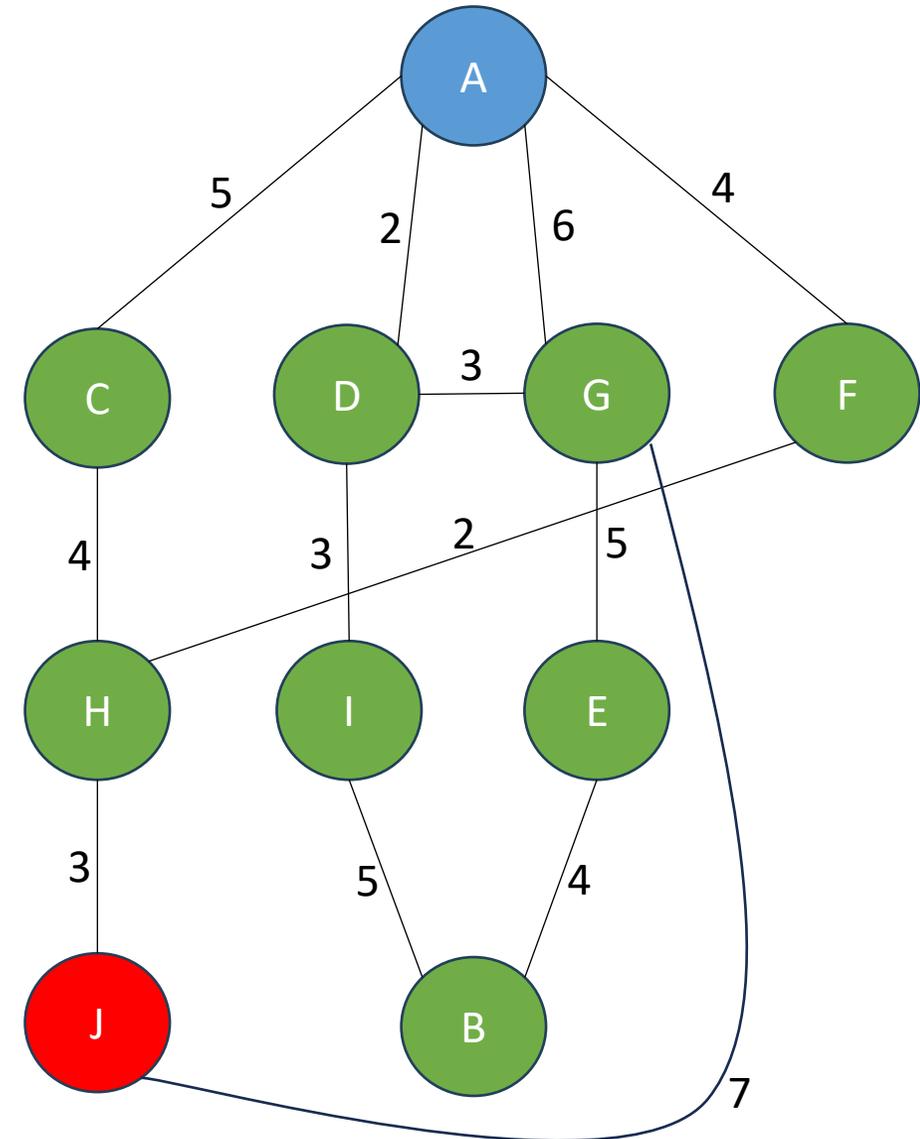
- **Búsqueda por horizonte:**
 - *D*: El algoritmo evalúa las posibles secuencias de movimientos dentro del horizonte (2 niveles).
 - Vecinos desde *D*: *A* (visitado), *G* (peso 3) e *I* (peso 3).
 - Se simulan secuencias:
 - $D \rightarrow I \rightarrow B$ (coste: 8)
 - $D \rightarrow G \rightarrow E$ (coste: 8)
 - $D \rightarrow G \rightarrow J$ (coste: 10)
 - Las secuencias $D \rightarrow I \rightarrow B$ y $D \rightarrow I \rightarrow E$ parecen la más prometedoras.
 - Se elige $D \rightarrow I \rightarrow B$ por estar antes en la cola.
 - *I*: El algoritmo evalúa las posibles secuencias de movimientos dentro del horizonte (2 niveles).
 - Vecinos desde *I*: *D* (visitado) y *B* (peso 5)
 - Se elige $I \rightarrow B \rightarrow E$ por ser la única posible.



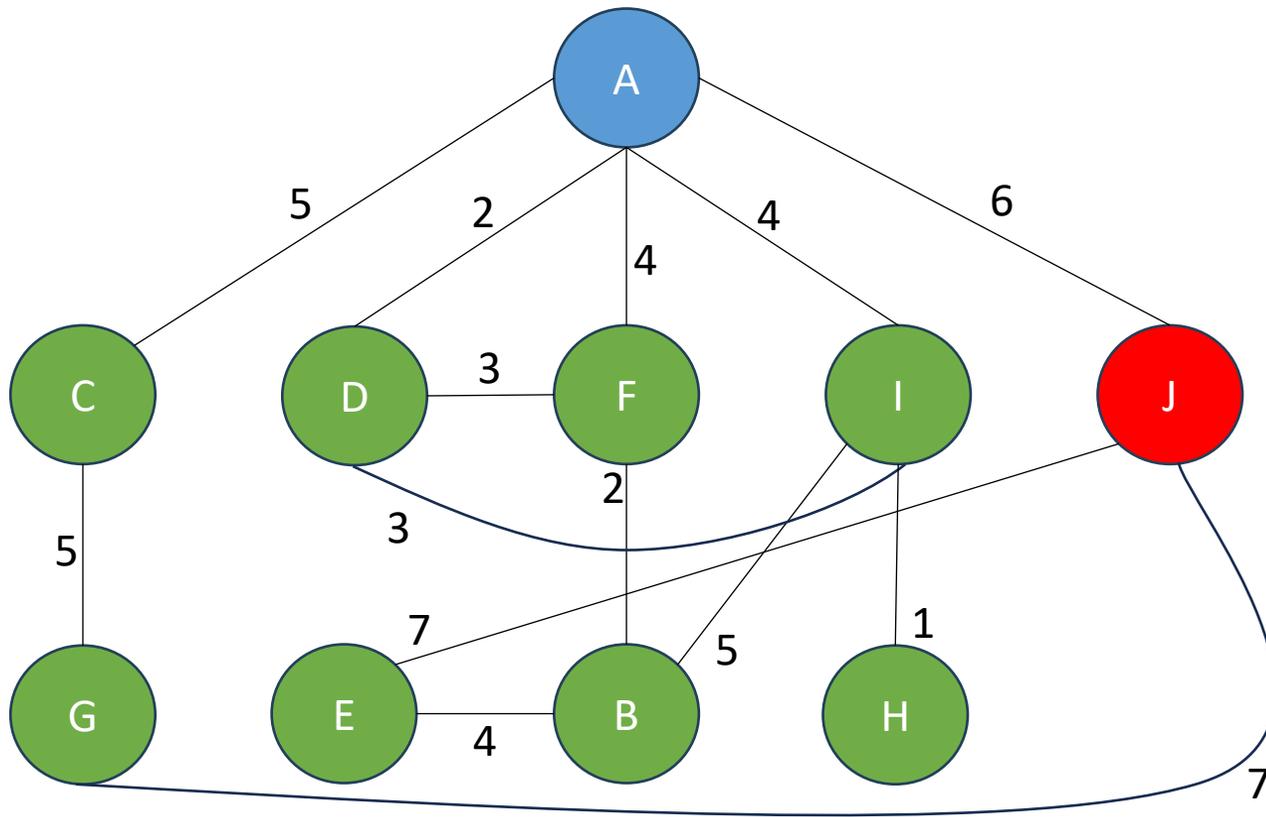
Ejercicio 3

- Búsqueda por horizonte:

- *B*: El algoritmo evalúa las posibles secuencias de movimientos dentro del horizonte (2 niveles).
 - Vecinos desde *B*: *I* (visitado) y *E* (peso 4)
 - Se elige $B \rightarrow E \rightarrow G$ por ser la única posible.
- *E*: El algoritmo evalúa las posibles secuencias de movimientos dentro del horizonte (2 niveles).
 - Vecinos desde *E*: *B* (visitado) y *G* (peso 5)
 - **Cabe destacar que *D* había sido visitado.**
 - Se elige $E \rightarrow G \rightarrow J$ por ser la única posible.
- *G*: El algoritmo evalúa las posibles secuencias de movimientos dentro del horizonte (2 niveles).
 - Vecinos desde *G*: *D* (visitado) y *J* (peso 7)
 - Se elige $G \rightarrow J$ por ser la única posible.
- *J*: Es el vértice objetivo.
- Camino recorrido: $A \rightarrow D \rightarrow I \rightarrow B \rightarrow E \rightarrow G \rightarrow J$
- Coste total: 26



Ejercicio 4

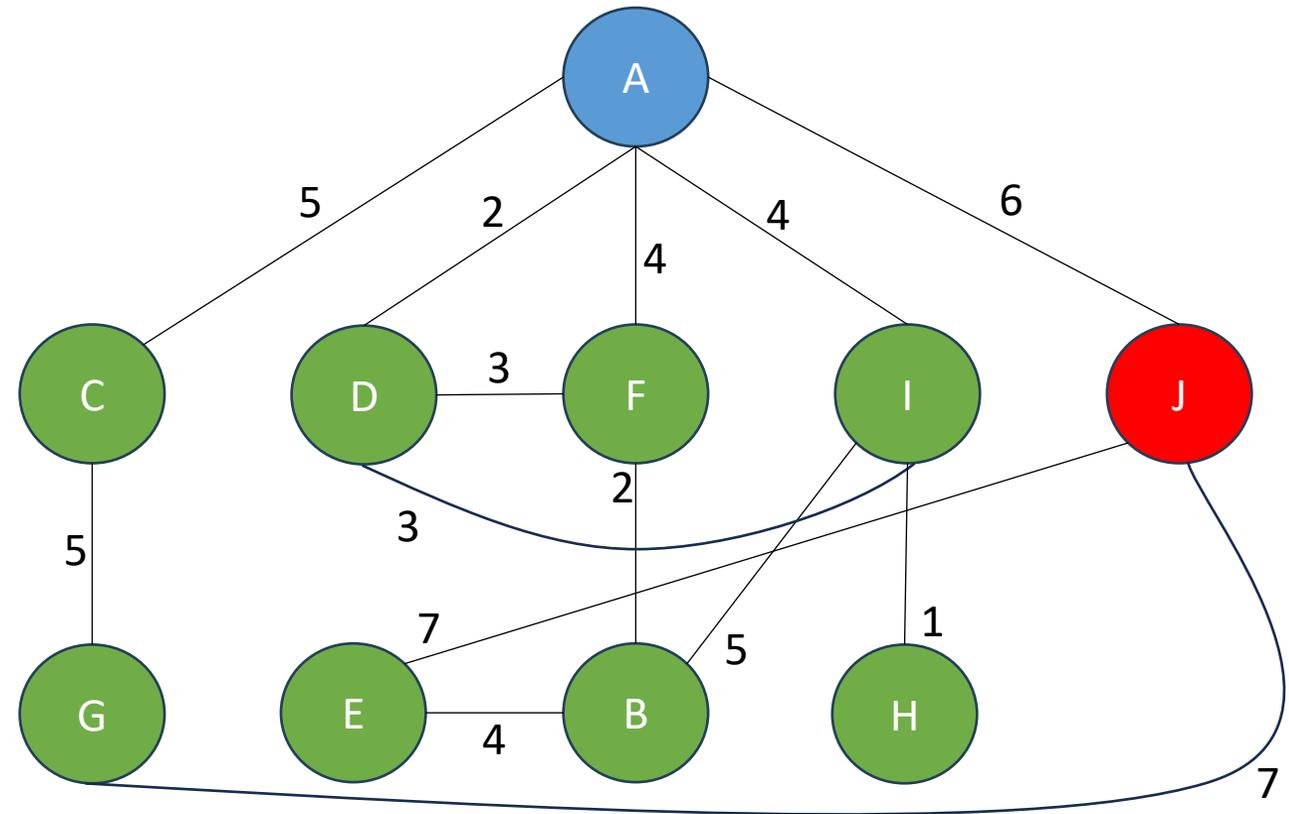


- Sobre el grafo anterior:
 - Siendo A el vértice de inicio.
 - Siendo J el vértice objetivo.
- Se pide aplicar:
 - Búsqueda por subobjetivos (D, I, J)
 - Búsqueda por ascenso de colinas
 - Búsqueda por horizonte (2)

Ejercicio 4

• Hill Climbing:

- Inicio en *A*:
 - Vecinos de *A*: *C* (peso 5), *D* (peso 2), *F* (peso 4), *I* (peso 4), *J* (peso 6).
 - Se selecciona *D*, que tiene el menor peso (2).
- Desde *D*:
 - Vecinos de *D*: *A* (visitado), *I* (peso 3), *F* (peso 3).
 - Se selecciona *F* (entra antes en la cola).
- Desde *F*:
 - Vecinos de *F*: *A* (visitado), *D* (visitado), *B* (peso 2).
 - Se selecciona *B* con un coste de 2.
- Desde *B*:
 - Vecinos de *B*: *E* (peso 4), *I* (peso 5), *F* (visitado).
 - Se selecciona *E*, el de menor coste, con coste 4.
- Desde *E*:
 - Vecinos de *E*: *B* (visitado), *J* (peso 7).
 - Se selecciona *J* con un coste de 7.



- Camino Final: $A \rightarrow D \rightarrow F \rightarrow B \rightarrow E \rightarrow J$
- Costo Total: $2+3+2+4+7=18$

Ejercicio 4

- Búsqueda por horizonte:

- Inicio en A : El algoritmo evalúa las posibles secuencias de movimientos dentro del horizonte (2 niveles).

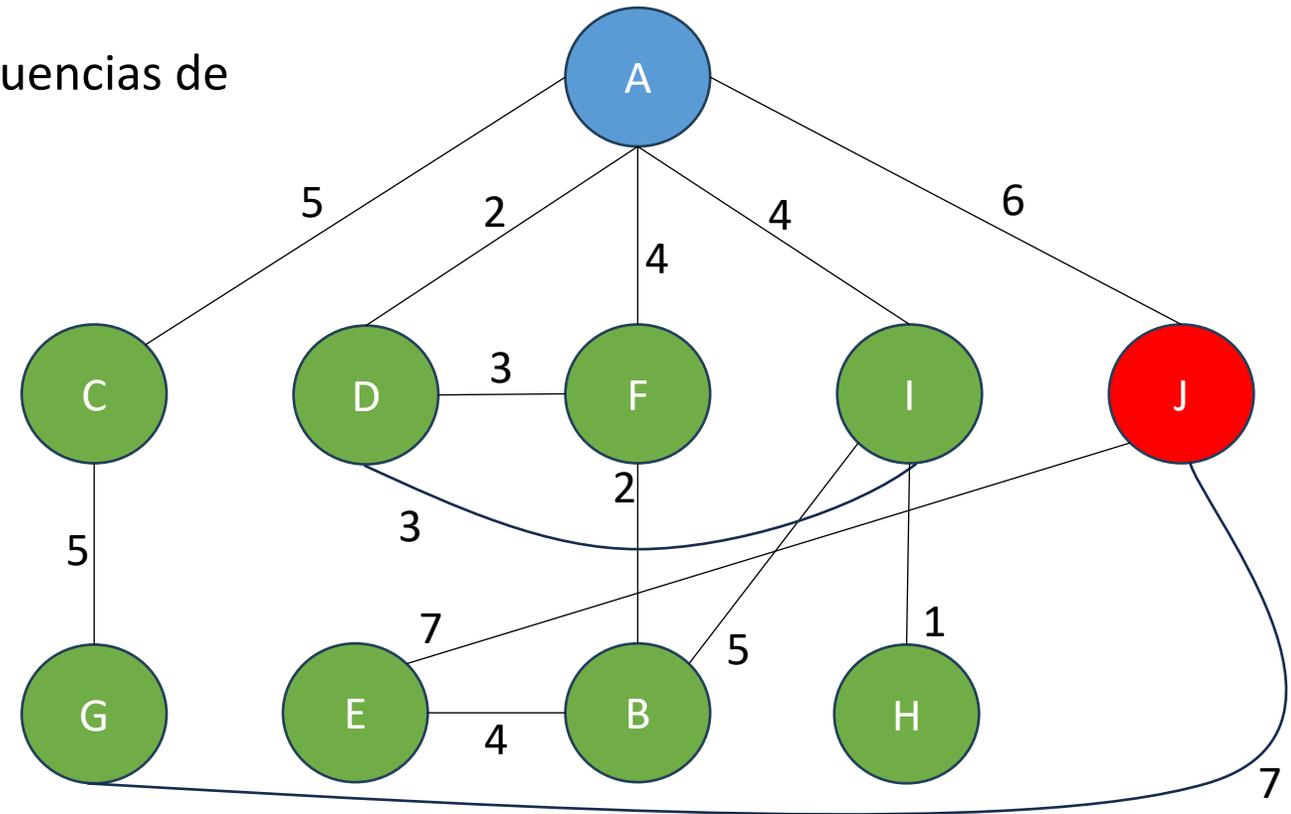
- Vecinos desde A : C, D, F, I, J

- Se simulan secuencias:

- $A \rightarrow C \rightarrow G$ (coste: 10)
- $A \rightarrow D \rightarrow F$ (coste: 5) (primero en la cola)
- $A \rightarrow D \rightarrow I$ (coste: 6)
- $A \rightarrow F \rightarrow B$ (coste: 6)
- $A \rightarrow F \rightarrow D$ (coste: 7)
- $A \rightarrow I \rightarrow B$ (coste: 9)
- $A \rightarrow I \rightarrow H$ (coste: 5)
- $A \rightarrow J \rightarrow G$ (coste 6)

- Las secuencias $A \rightarrow D \rightarrow F$ y $A \rightarrow I \rightarrow H$ parecen las más prometedoras.

- Se elige $A \rightarrow D \rightarrow F$ por estar antes en la cola.



Ejercicio 4

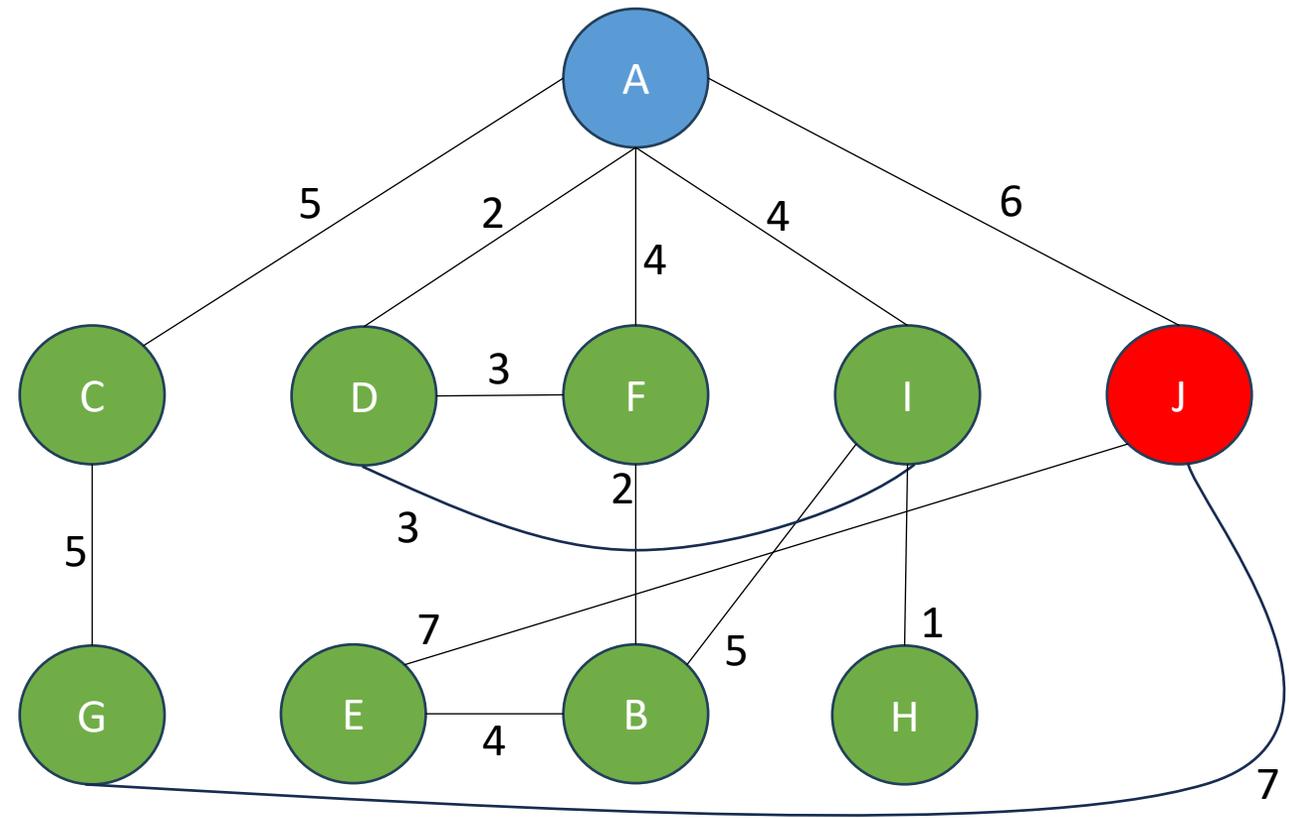
- Búsqueda por horizonte:

- *D*:

- Vecinos de *D*: *A* (visitado), *F*, *I*
- Secuencias posibles:
 - $D \rightarrow F \rightarrow B$ (coste: 5)
 - $D \rightarrow I \rightarrow B$ (coste: 8)
 - $D \rightarrow I \rightarrow H$ (coste: 4)
- La secuencia seleccionada es $D \rightarrow I \rightarrow H$ (coste: 4).

- *I*:

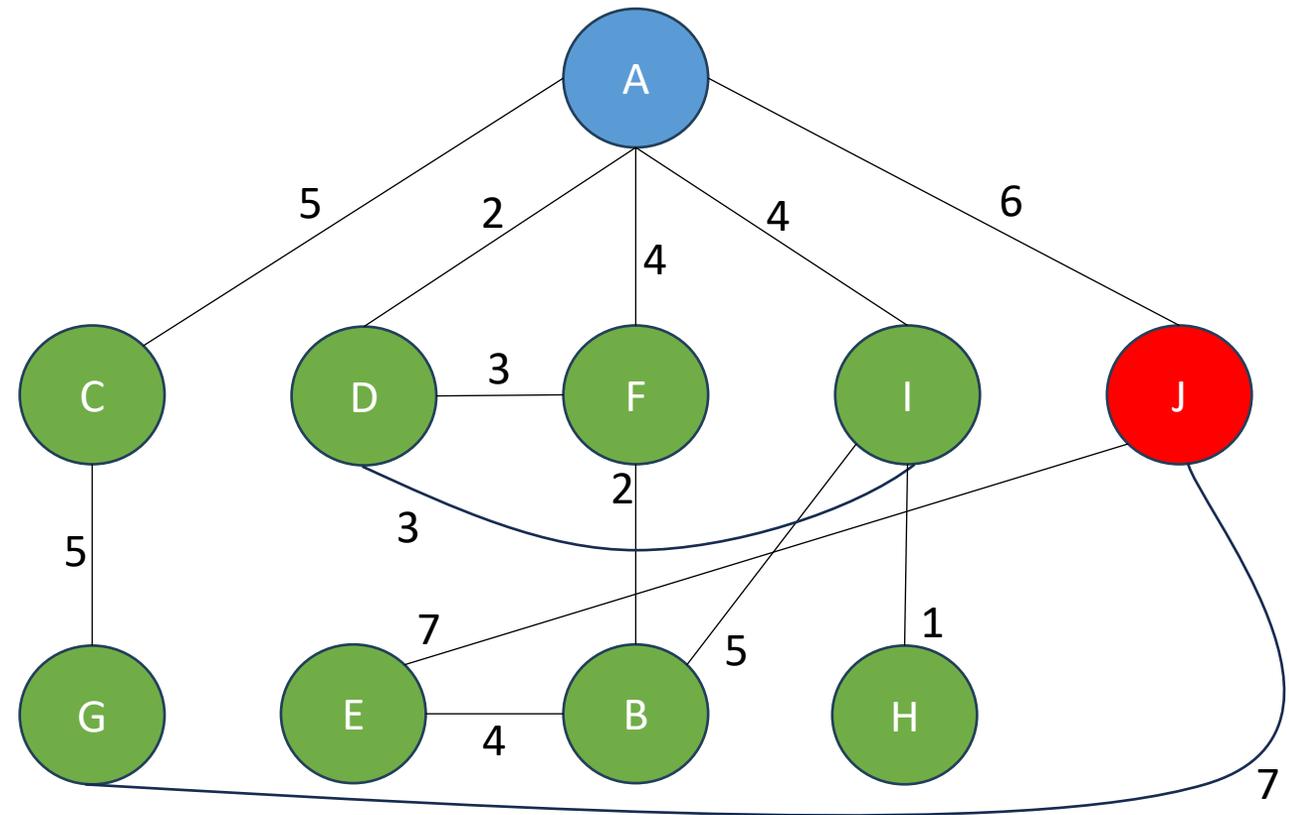
- Vecinos de *I*: *A* (visitado), *B*, *D* (visitado), *H*.
- Secuencias posibles
 - $I \rightarrow B \rightarrow E$ (coste: 9)
 - $I \rightarrow B \rightarrow F$ (coste: 7)
 - $I \rightarrow H$ (coste: 1, pero no puede seguir por este camino)
- La secuencia seleccionada es $I \rightarrow B \rightarrow F$ (coste: 6).



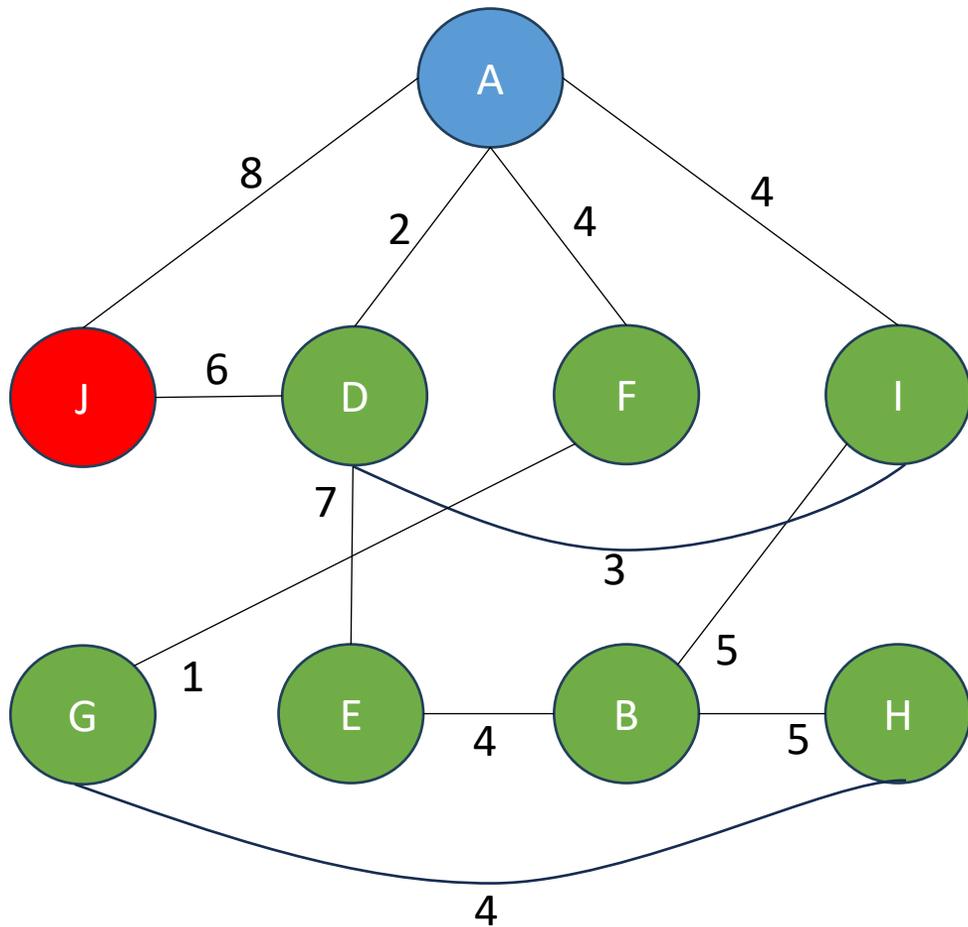
Ejercicio 4

- Búsqueda por horizonte:

- *B*:
 - Vecinos de *B*: *E*, *F*, *I* (visitado)
 - Secuencias posibles:
 - $B \rightarrow E \rightarrow J$ (coste: 11)
 - **Cabe recordar que *A* y *D* han sido visitados.**
 - La secuencia seleccionada es $B \rightarrow E \rightarrow J$ (coste: 11).
- *J* es el objetivo.
- Camino recorrido: $A \rightarrow D \rightarrow I \rightarrow B \rightarrow E \rightarrow J$
- Coste total: 21



Ejercicio 5

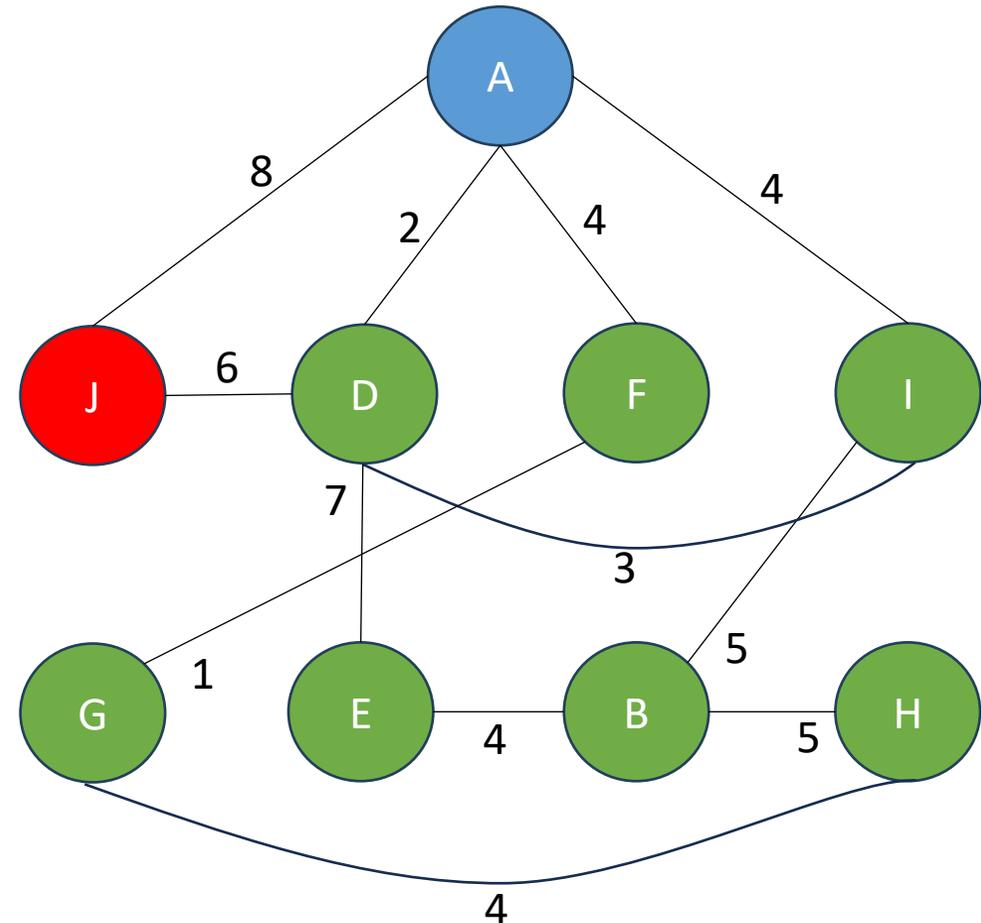


- Sobre el grafo anterior:
 - Siendo A el vértice de inicio.
 - Siendo J el vértice objetivo.
- Se pide aplicar:
 - Búsqueda por subobjetivos (D, B, J)
 - Búsqueda por ascenso de colinas
 - Búsqueda por horizonte (2)

Ejercicio 5

• Hill Climbing:

- Inicio en *A*:
 - Vecinos: *D* (peso 2), *F* (peso 4), *I* (peso 4), *J* (peso 8).
 - Se selecciona *D* con un coste de 2.
- Desde *D*:
 - Vecinos de *D*: *A* (visitado), *I* (peso 3), *E* (peso 7).
 - Se selecciona *I* (menor coste).
- Desde *I*:
 - Vecinos de *I*: *A* (visitado), *D* (visitado), *B* (peso 5)
 - Se selecciona *B* como única posible opción.
- Desde *B*:
 - Vecinos desde *B*: *I* (visitado), *E* (peso 4), *H* (peso 5)
 - Se selecciona *E* por ser el de menor coste.
- Desde *E*:
 - Vecinos desde *E*: *B* (visitado) y *D* (visitado)
 - No tiene opciones válidas.
- Se queda atrapado y no puede seguir.



Ejercicio 5

- Búsqueda por horizonte:

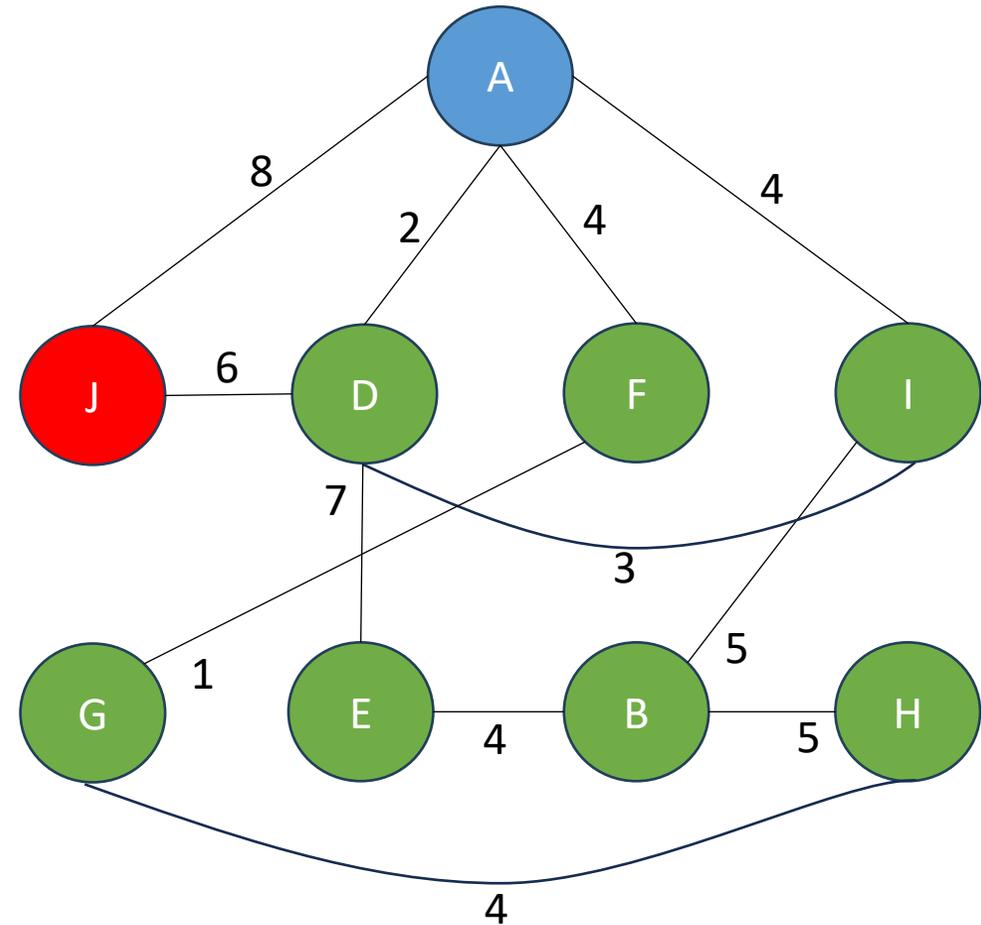
- A:

- Vecinos de A: *D, F, I, J*
- Secuencias posibles:
 - $A \rightarrow D \rightarrow J$ (coste: 8)
 - $A \rightarrow D \rightarrow E$ (coste: 9)
 - $A \rightarrow D \rightarrow I$ (coste: 5) (primera en la cola)
 - $A \rightarrow F \rightarrow G$ (coste: 5)
 - $A \rightarrow I \rightarrow V$ (coste: 9)
 - $A \rightarrow J \rightarrow D$ (coste: 14)
- La secuencia seleccionada es $A \rightarrow D \rightarrow I$ (coste: 5).

- D:

- Vecinos de D: *J, E, I*
- Secuencias posibles:
 - $D \rightarrow E \rightarrow B$ (coste: 11)
 - $D \rightarrow I \rightarrow B$ (coste: 8)
 - $D \rightarrow J \rightarrow A$ (coste: 8) (A ya sido visitado)

- ¿Es posible encontrar alguna solución en este punto?



DESARROLLO DE JUEGOS CON INTELIGENCIA ARTIFICIAL

Búsqueda en el espacio de estados y agentes

Tema 3. Búsqueda con heurísticas débiles

©2023 Autores
David María Arribas,
Sergio Hernández García,
Nicolás H. Rodríguez Uribe,

Algunos derechos reservados
Este documento se distribuye bajo la licencia
"Atribución-CompartirIgual 4.0 Internacional" de Creative Commons, disponible en
<https://creativecommons.org/licenses/by-sa/4.0/deed.es>



Nicolás Rodríguez
nicolas.rodriguez@urjc.es

David María
david.maria@urjc.es

Sergio Hernández
sergio.Hernandez@urjc.es



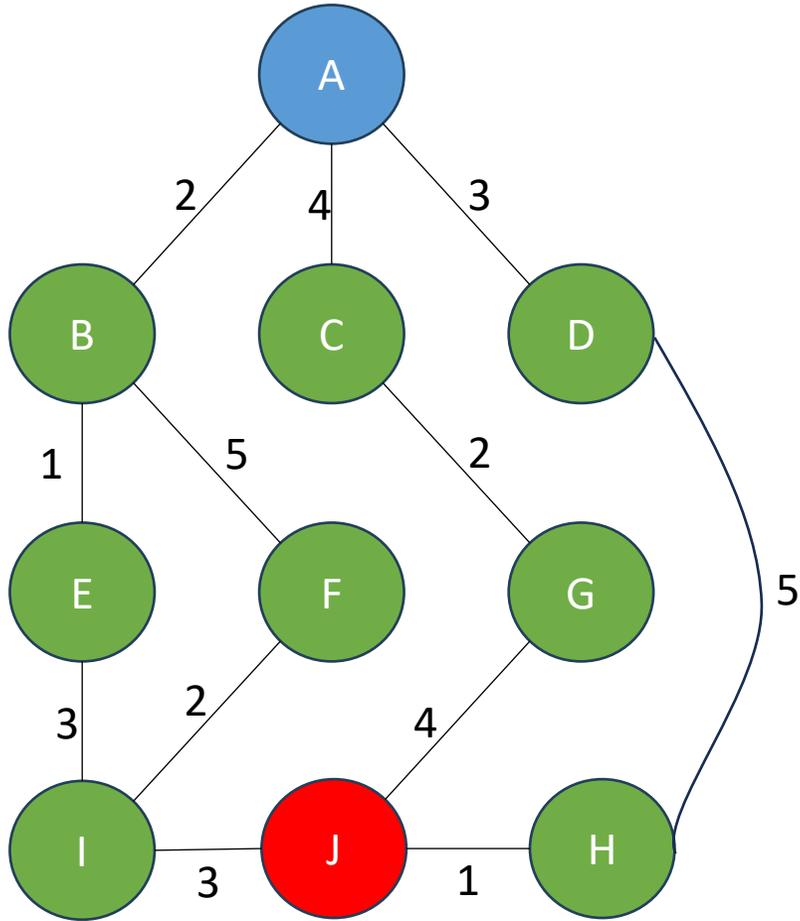
DESARROLLO DE JUEGOS CON INTELIGENCIA ARTIFICIAL

Búsqueda en el espacio de estados y agentes

Tema 4. Búsqueda con heurísticas fuertes



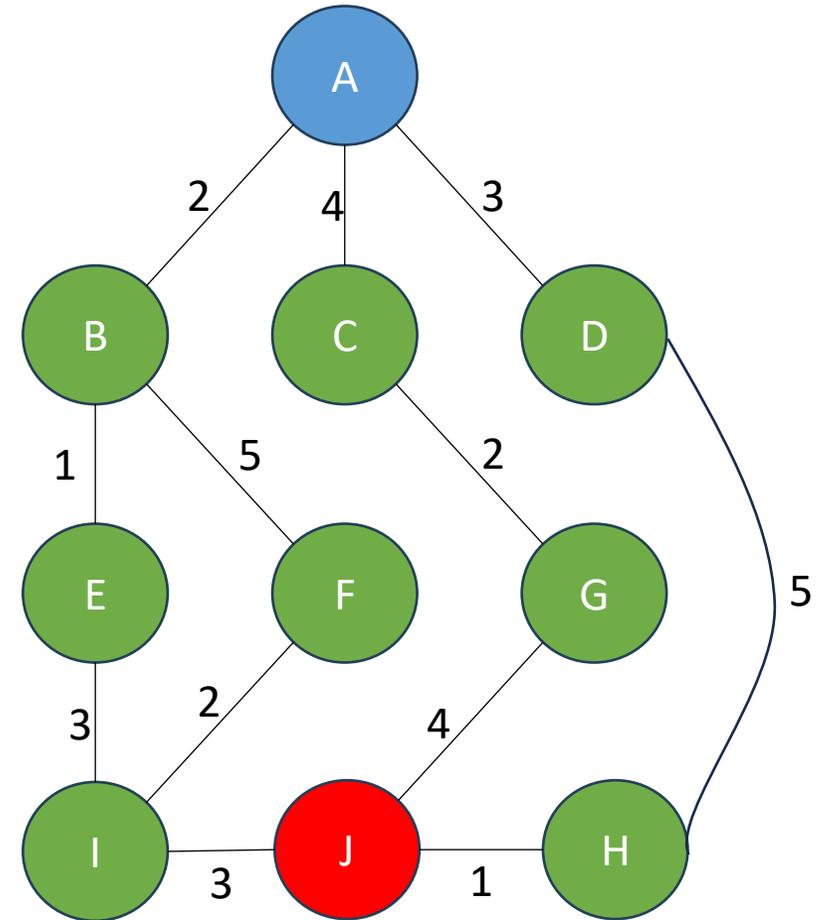
Ejercicio 1



- Sobre el grafo anterior:
 - Siendo A el vértice de inicio.
 - Siendo J el vértice objetivo.
- Se pide aplicar:
 - A*

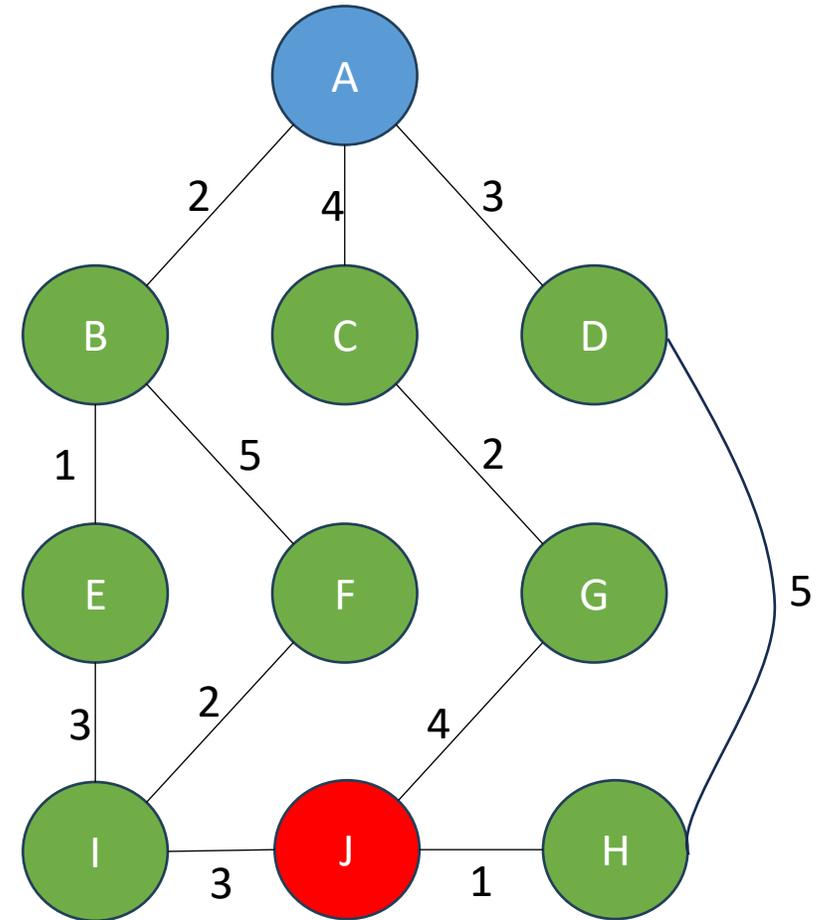
Ejercicio 1

- **A: 4 (a través de $B \rightarrow E \rightarrow I \rightarrow J$)**
 - Desde A , el camino estimado más corto a J pasa por B , luego E , seguido de I , hasta llegar a J , con un coste estimado total de 4 pasos.
- **B: 3 (a través de $E \rightarrow I \rightarrow J$)**
 - Desde B , el camino más corto estimado hacia J es pasando por E , luego I , y finalmente J , con un coste estimado de 3 pasos.
- **C: 3 (a través de $G \rightarrow J$)**
 - Desde C , el camino más corto estimado a J es pasando por G y luego directamente a J , con un coste estimado de 3.
- **D: 4 (a través de $H \rightarrow J$)**
 - Desde D , el camino más corto estimado hacia J pasa por H , que está directamente conectado a J , con un coste estimado de 4.



Ejercicio 1

- **E: 2 (a través de $I \rightarrow J$)**
 - Desde E , el camino más corto estimado hacia J es a través de I , que está conectado a J , con un coste estimado de 2.
- **F: 2 (a través de $I \rightarrow J$)**
 - Desde F , el camino más corto estimado hacia J también pasa por I , con un coste estimado de 2.
- **G: 1 (conexión directa a J)**
- **H: 1 (conexión directa a J)**
- **I: 1 (conexión directa a J)**
 - Desde G, H, I el nodo J está directamente conectado con un coste de 1.
- **J: 0 (nodo objetivo)**
 - J es el nodo objetivo, por lo que la heurística para él es 0.



Ejercicio 1

- Desde A :

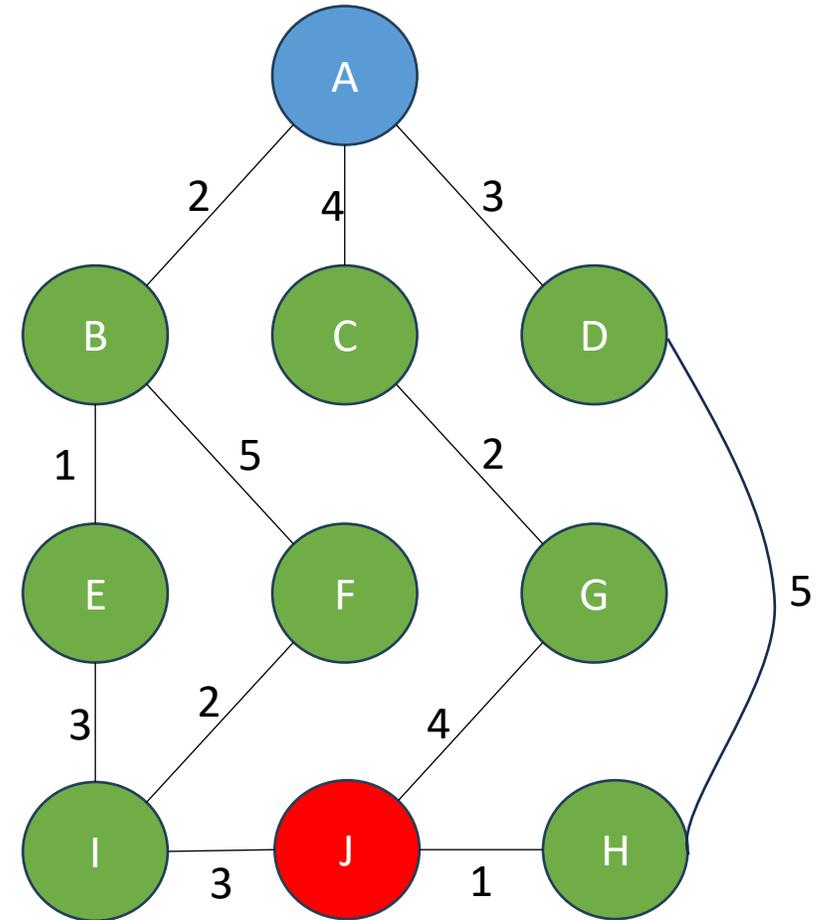
- $A \rightarrow B$: $g(B)=2, h(B)=3, f(B)=2+3=5$
- $A \rightarrow C$: $g(C)=4, h(C)=3, f(C)=4+3=7$
- $A \rightarrow D$: $g(D)=3, h(D)=4, f(D)=3+4=7$
- Nodo seleccionado: B con $f(B)=5$ (el menor valor de $f(n)$).

- Desde B :

- $B \rightarrow A$ (ya visitado)
- $B \rightarrow E$: $g(E)=g(B)+1=2+1=3, h(E)=2, f(E)=3+2=5$
- $B \rightarrow F$: $g(F)=g(B)+5=2+5=7, h(F)=2, f(F)=7+2=9$
- Nodo seleccionado: E con $f(E)=5$.

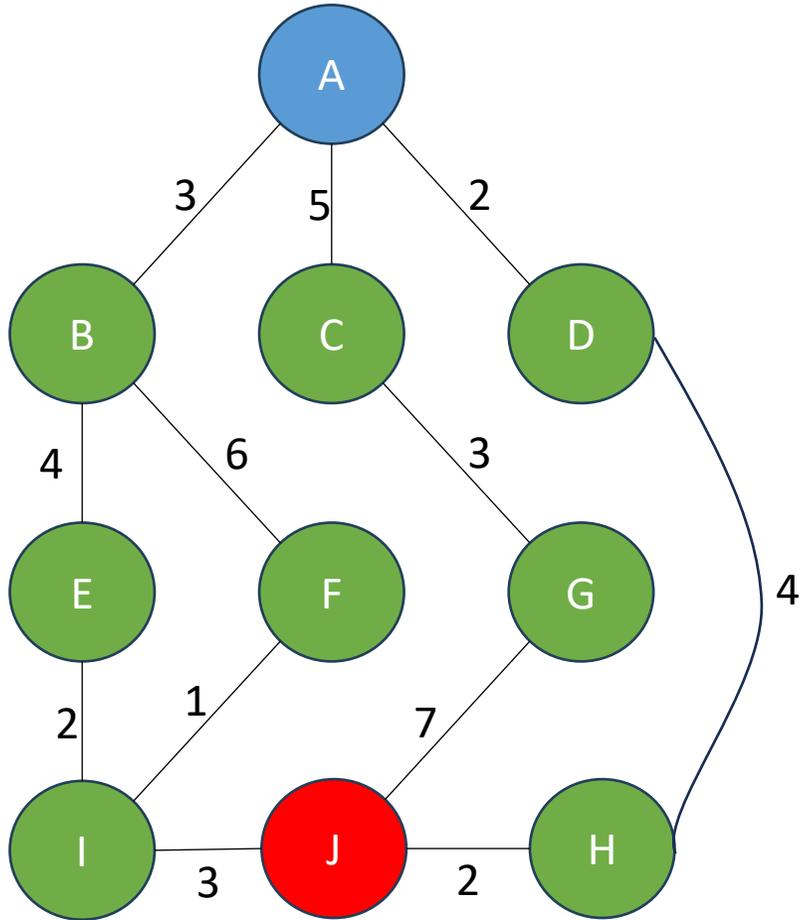
- Desde I :

- $I \rightarrow E$ (ya visitado)
- $I \rightarrow F$ (ya visitado)
- $I \rightarrow J$: $g(J)=g(I)+3=6+3=9, h(J)=0, f(J)=9+0=9$
- Nodo seleccionado: J con $f(J)=9$.



- Camino final: $A \rightarrow B \rightarrow E \rightarrow I \rightarrow J$
- Coste total: $2+1+3+3=9$

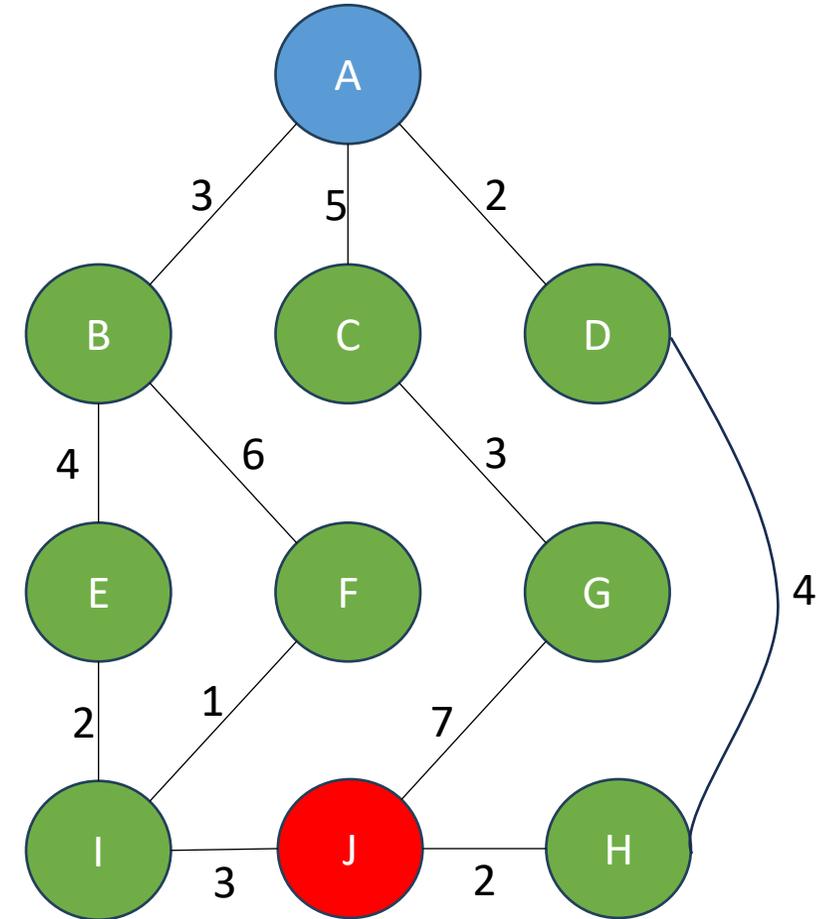
Ejercicio 2



- Sobre el grafo anterior:
 - Siendo A el vértice de inicio.
 - Siendo J el vértice objetivo.
- Se pide aplicar:
 - A*

Ejercicio 2

- A : 3 (a través de $D \rightarrow H \rightarrow J$)
- B : 4 (a través de $E \rightarrow I \rightarrow J$)
- C : 3 (a través de $G \rightarrow J$)
- D : 2 (a través de $H \rightarrow J$)
- E : 3 (a través de $I \rightarrow J$)
- F : 3 (a través de $I \rightarrow J$)
- G : 1 (conexión directa a J)
- H : 1 (conexión directa a J)
- I : 1 (conexión directa a J)
- J : 0 (es el nodo objetivo)



Ejercicio 2

- Desde A

- $A \rightarrow B$: $g(B)=3$, $h(B)=4$, $f(B)=3+4=7$
- $A \rightarrow C$: $g(C)=5$, $h(C)=3$, $f(C)=5+3=8$
- $A \rightarrow D$: $g(D)=2$, $h(D)=2$, $f(D)=2+2=4$
- Nodo seleccionado: D con $f(D)=4$.

- Desde D :

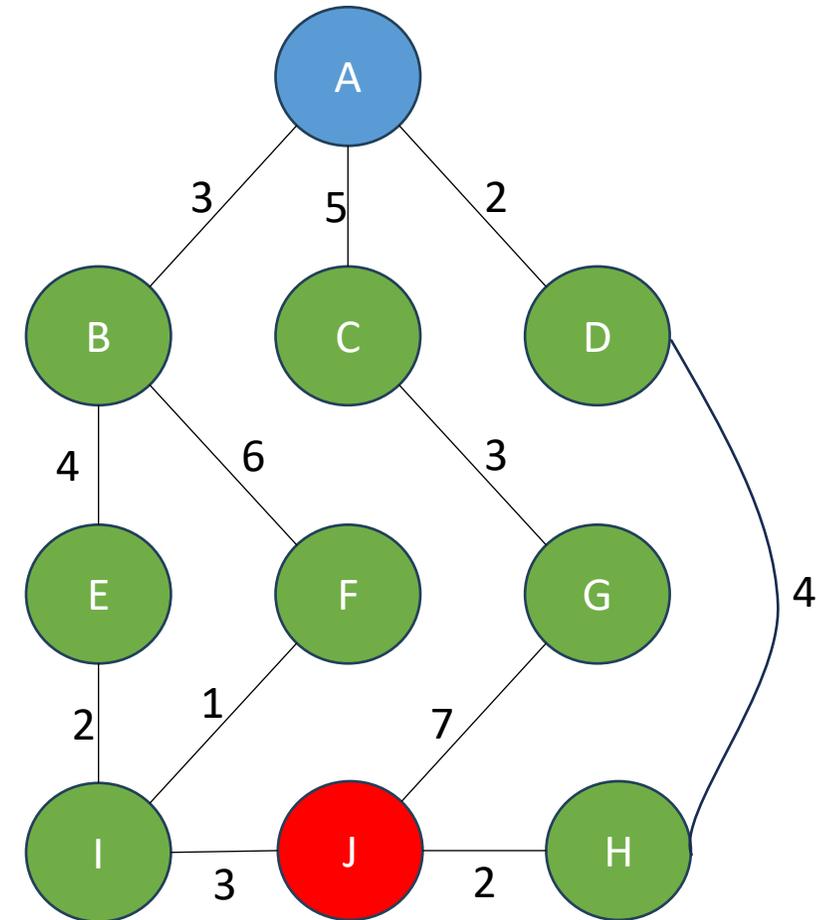
- $D \rightarrow A$ (ya visitado)
- $D \rightarrow H$: $g(H)=g(D)+4=2+4=6$, $h(H)=1$, $f(H)=6+1=7$
- Nodo seleccionado: H con $f(H)=7$.

- Desde H :

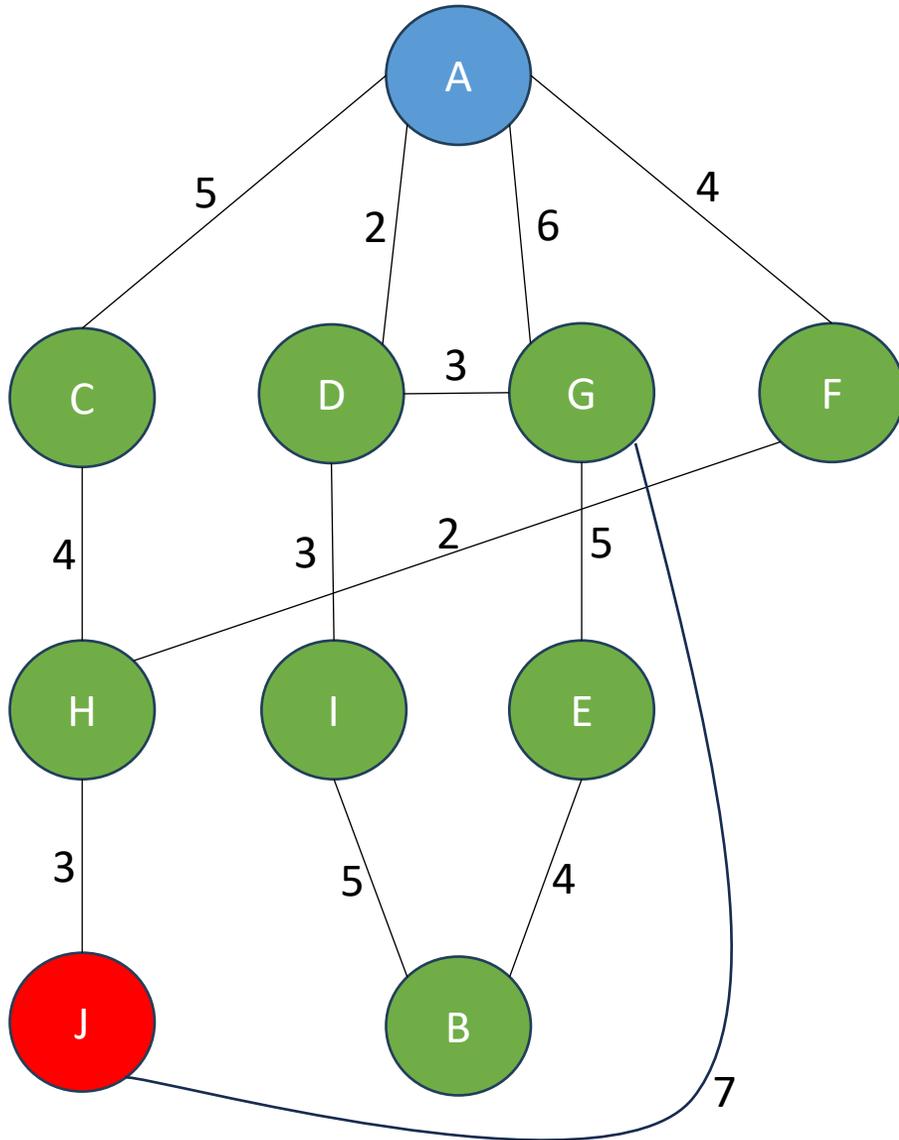
- $H \rightarrow D$ (ya visitado)
- $H \rightarrow J$: $g(J)=g(H)+2=6+2=8$, $h(J)=0$, $f(J)=8+0=8$
- Nodo seleccionado: J con $f(J)=8$.

- Camino final: $A \rightarrow D \rightarrow H \rightarrow J$

- Coste total: $2+4+2=8$



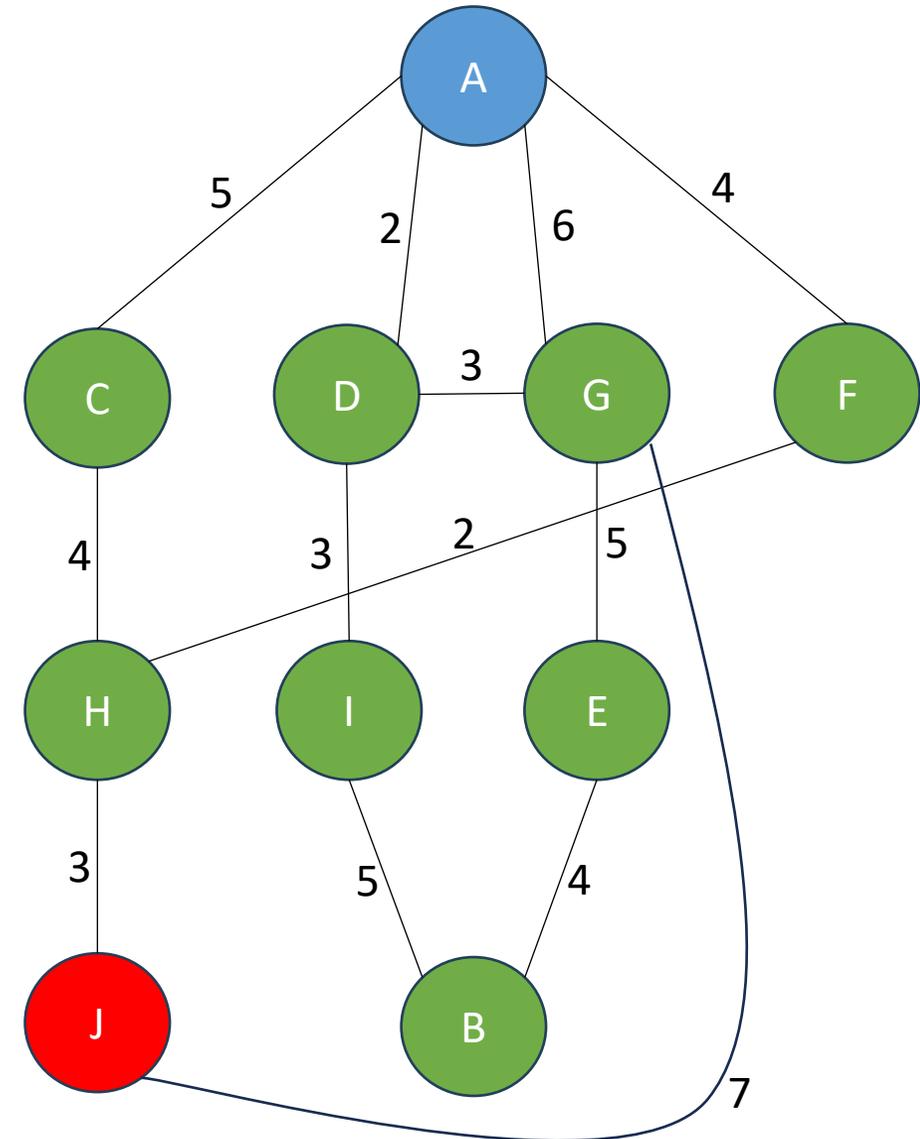
Ejercicio 3



- Sobre el grafo anterior:
 - Siendo A el vértice de inicio.
 - Siendo J el vértice objetivo.
- Se pide aplicar:
 - A*

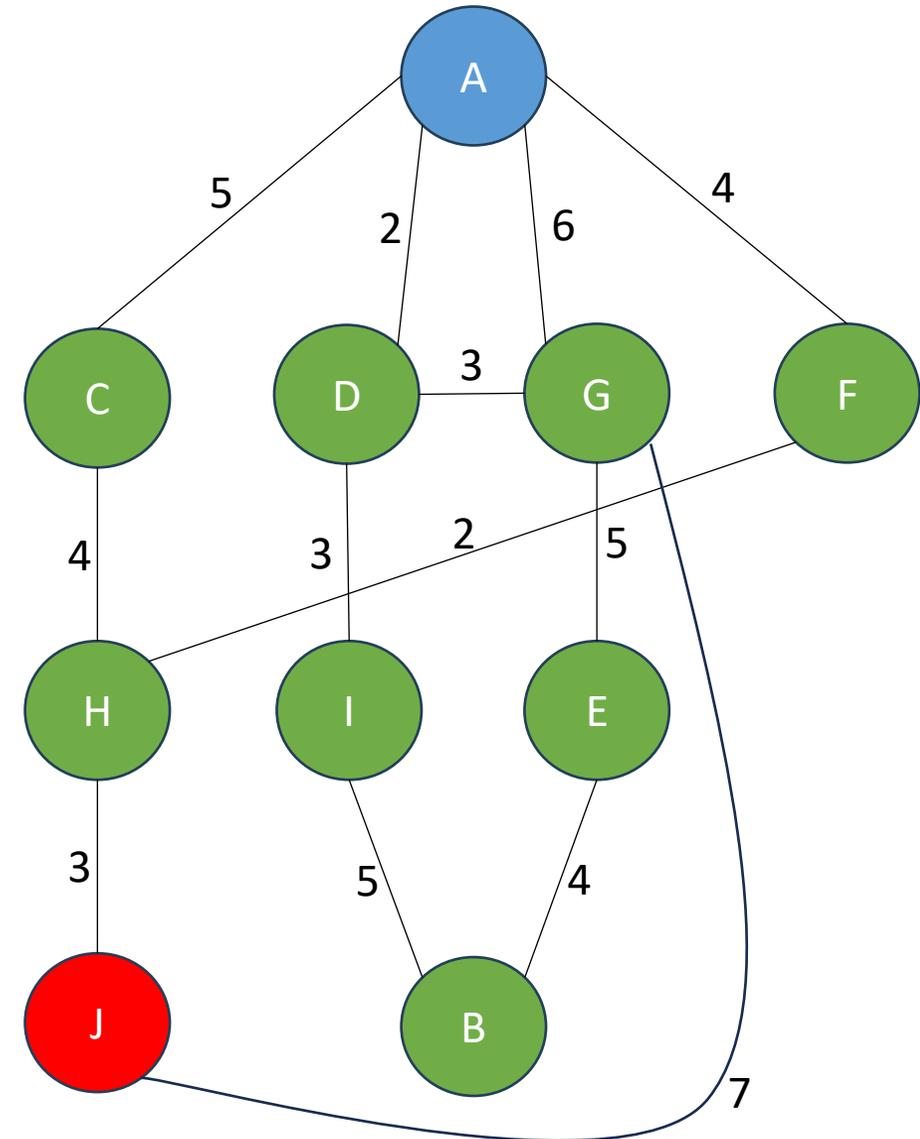
Ejercicio 3

- **A: 6**
 - A está a tres aristas de J, pasando por $D \rightarrow G \rightarrow J$ o $F \rightarrow H \rightarrow J$, lo cual es un camino más largo en términos de peso acumulado.
 - Heurística $h(A)=6$ (el camino estimado más corto en términos de peso).
- **B: 5**
 - Desde B, el camino más corto hacia J es pasando por $E \rightarrow G \rightarrow J$ o por $I \rightarrow H \rightarrow J$.
 - Heurística $h(B)=5$ (estimación del camino más corto).
- **C: 7**
 - Desde C, el camino hacia J es pasando por $H \rightarrow J$, con un coste total de 7 (arista $C \rightarrow H, H \rightarrow J$).
 - Heurística $h(C)=7h$.



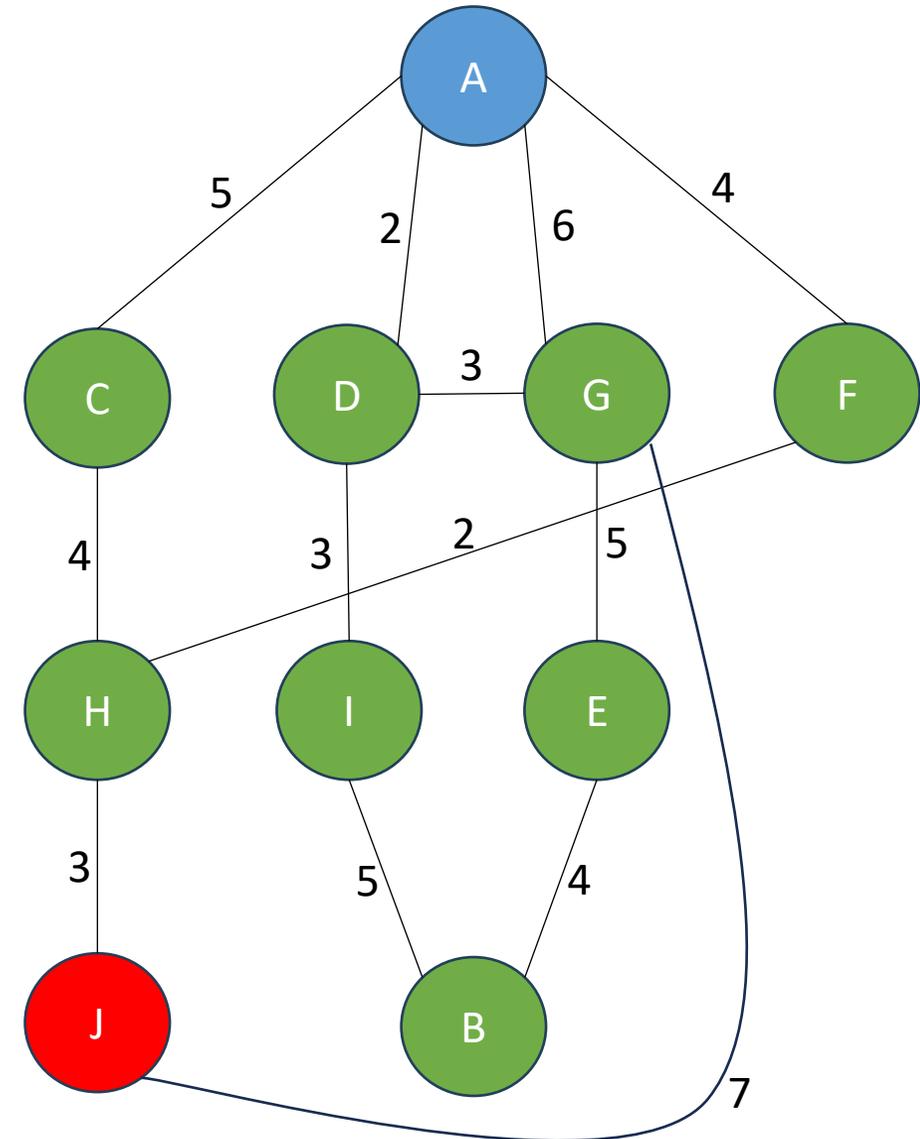
Ejercicio 3

- ***D*: 5**
 - Desde *D*, el camino más corto hacia *J* es a través de $G \rightarrow J$ o $I \rightarrow H \rightarrow J$.
 - Heurística $h(D)=5$
- ***E*: 4**
 - Desde *E*, el camino más corto hacia *J* es a través de $G \rightarrow J$.
 - Heurística $h(E)=4$.
- ***F*: 3**
 - Desde *F*, el camino más corto hacia *J* es a través de $H \rightarrow J$, con un coste de 3 (arista $F \rightarrow H, H \rightarrow J$).
 - Heurística $h(F)=3$.



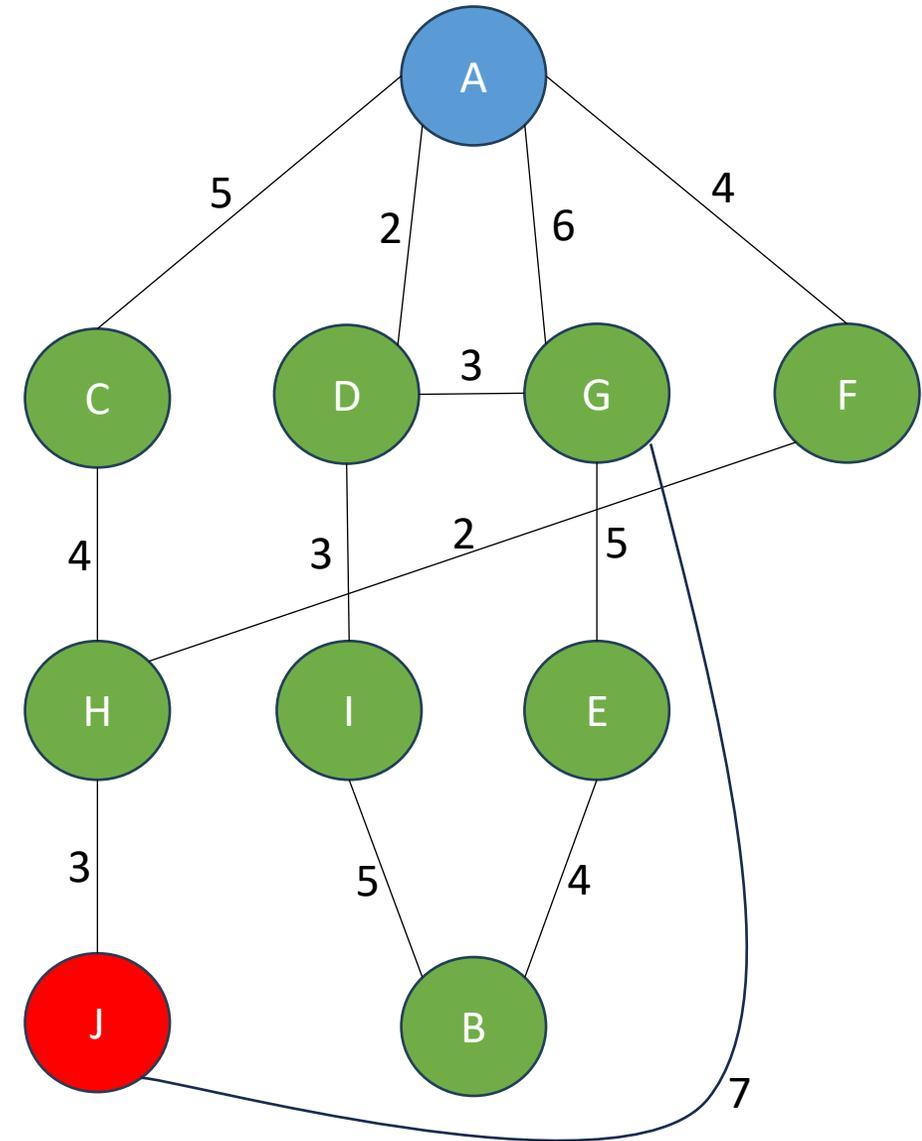
Ejercicio 3

- **$G: 1$**
 - G está directamente conectado a J , con un coste de 7.
 - Heurística $h(G)=1$.
- **$H: 2$**
 - H está directamente conectado a J , con un coste de 3.
 - Heurística $h(H)=2$.
- **$I: 6$**
 - Desde I , el camino más corto hacia J es a través de $H \rightarrow J$, lo cual implica un coste mayor debido a su posición.
 - Heurística $h(I)=6$.
- **$J: 0$**
 - Es el nodo objetivo, por lo que su heurística es 0.
 - Heurística $h(J)=0$.



Ejercicio 3

- $A: 6$
- $B: 5$
- $C: 7$
- $D: 5$
- $E: 4$
- $F: 3$
- $G: 1$
- $H: 2$
- $I: 6$
- $J: 0$ (el nodo objetivo)



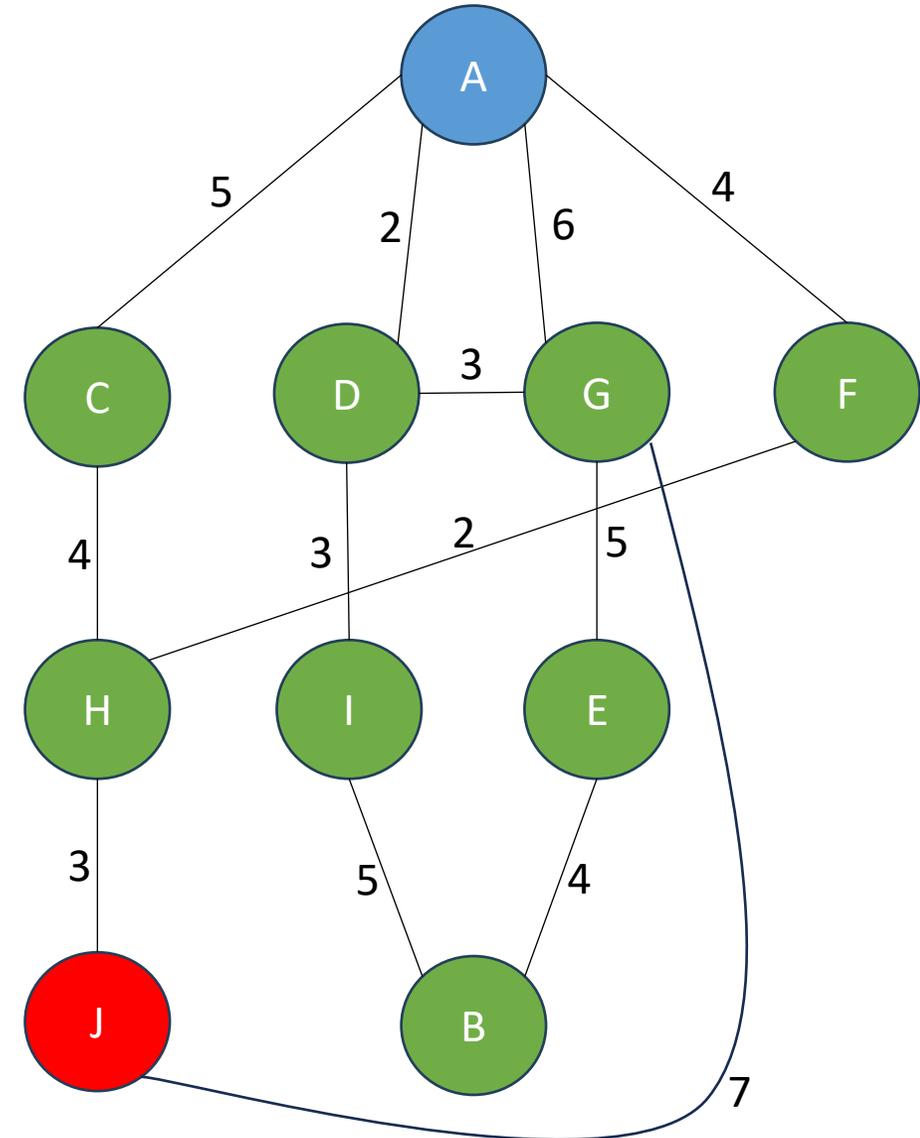
Ejercicio 3

- Desde A :

- $A \rightarrow C$: $g(C)=5, h(C)=7, f(C)=5+7=12$
- $A \rightarrow D$: $g(D)=2, h(D)=5, f(D)=2+5=7$
- $A \rightarrow G$: $g(G)=6, h(G)=1, f(G)=6+1=7$
- $A \rightarrow F$: $g(F)=4, h(F)=3, f(F)=4+3=7$
- Seleccionamos el nodo con el menor $f(n)$.
- Tenemos un empate entre $D, G,$ y F con $f=7$.
- Según el orden lexicográfico, seleccionamos D .

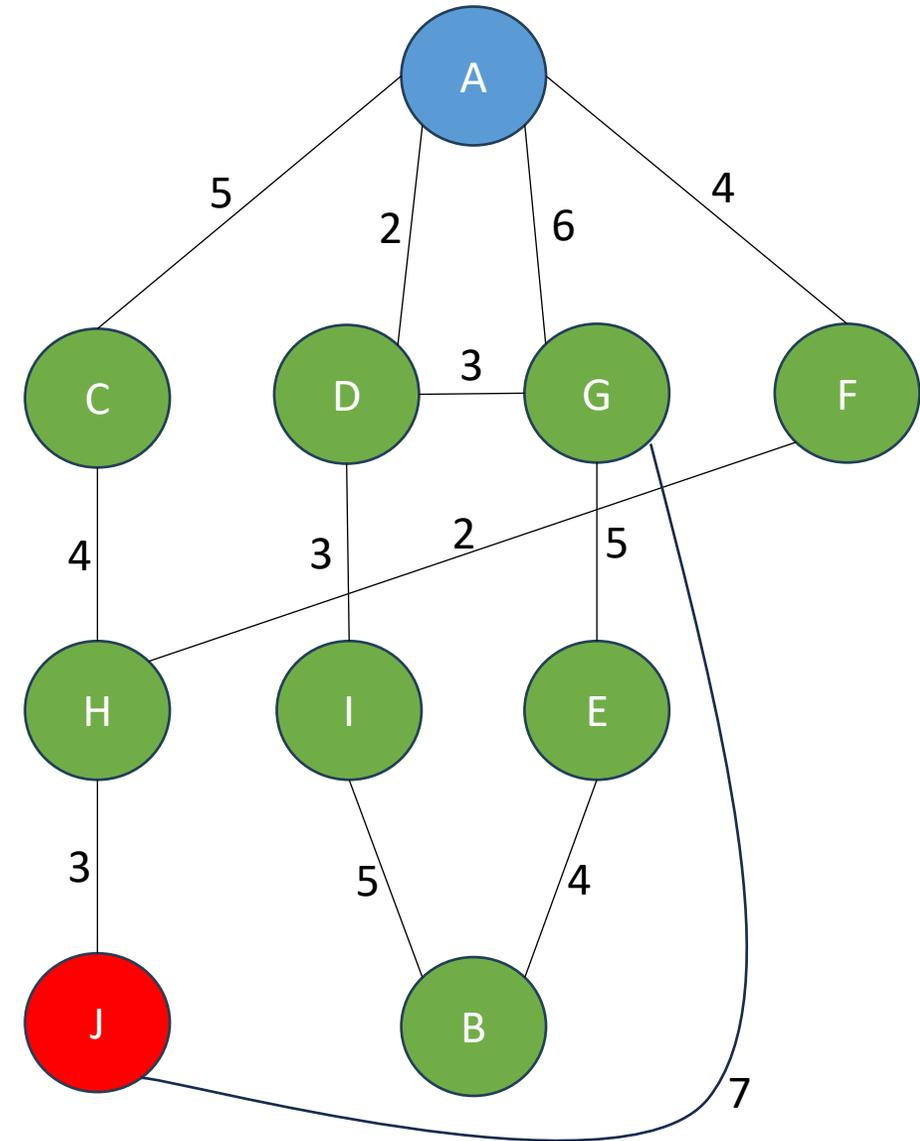
- Desde D :

- $D \rightarrow A$ (ya visitado, no se considera)
- $D \rightarrow G$: $g(G)=g(D)+3=2+3=5, h(G)=1, f(G)=5+1=6$
- $D \rightarrow I$: $g(I)=g(D)+3=2+3=5, h(I)=6, f(I)=5+6=11$
- Seleccionamos el nodo con el menor $f(n)$: G con $f(G)=6$.

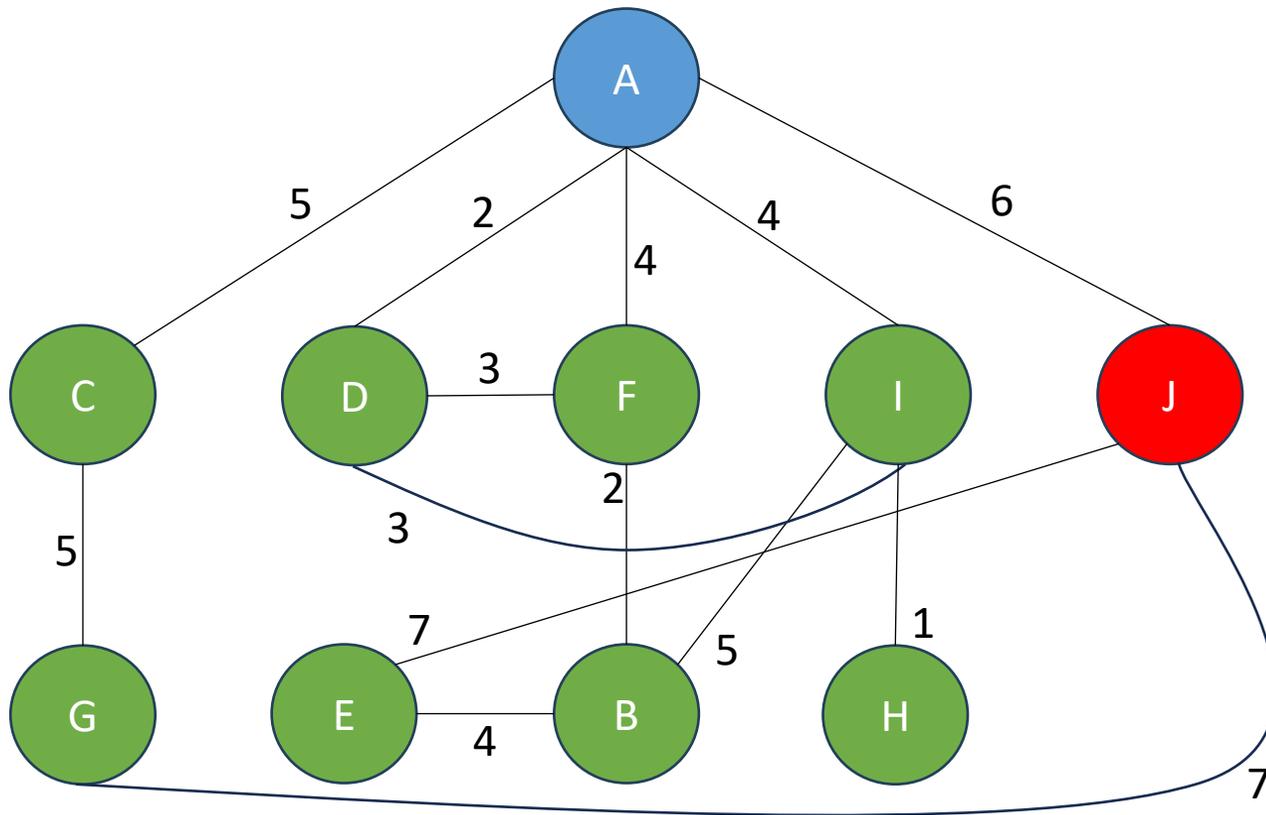


Ejercicio 3

- Desde G :
 - $G \rightarrow A$ (ya visitado, no se considera)
 - $G \rightarrow D$ (ya visitado, no se considera)
 - $G \rightarrow E$: $g(E) = g(G) + 5 = 5 + 5 = 10$, $h(E) = 4$, $f(E) = 10 + 4 = 14$
 - $G \rightarrow J$: $g(J) = g(G) + 7 = 5 + 7 = 12$, $h(J) = 0$, $f(J) = 12 + 0 = 12$
 - Seleccionamos el nodo con el menor $f(n)$: J con $f(J) = 12$.
- Camino Final: $A \rightarrow D \rightarrow G \rightarrow J$
- Coste total: $2 + 3 + 7 = 12$



Ejercicio 4



- Sobre el grafo anterior:
 - Siendo A el vértice de inicio.
 - Siendo J el vértice objetivo.
- Se pide aplicar:
 - A*

Ejercicio 4

- **Nodo A:**

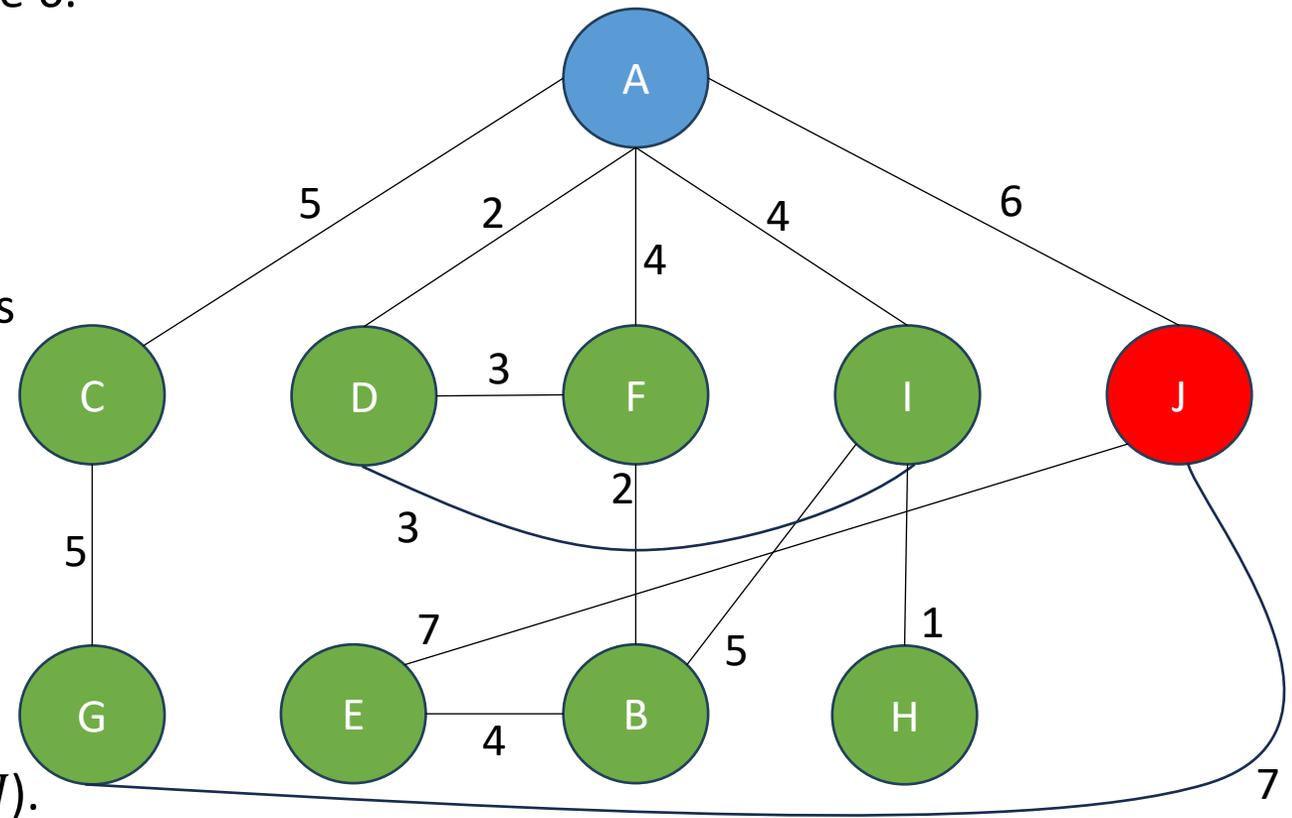
- Está directamente conectado a *J* con un peso de 6.
- Heurística $h(A) = 6$ (directamente el coste del camino más corto hacia *J*).

- **Nodo B:**

- Para llegar a *J* desde *B*, la mejor ruta es a través de *E* con un coste total de 4 (de *B* a *E*) + 7 (de *E* a *J*).
- Heurística $h(B) = 4$ (estimación más cercana a través de *E*).

- **Nodo C:**

- Desde *C*, el camino más cercano es pasando por *G*, con un coste de 5 (de *C* a *G*) + 7 (de *G* a *J*).
- Heurística $h(C) = 7$ (estimación más cercana a través de *G*).



Ejercicio 4

- **Nodo D :**

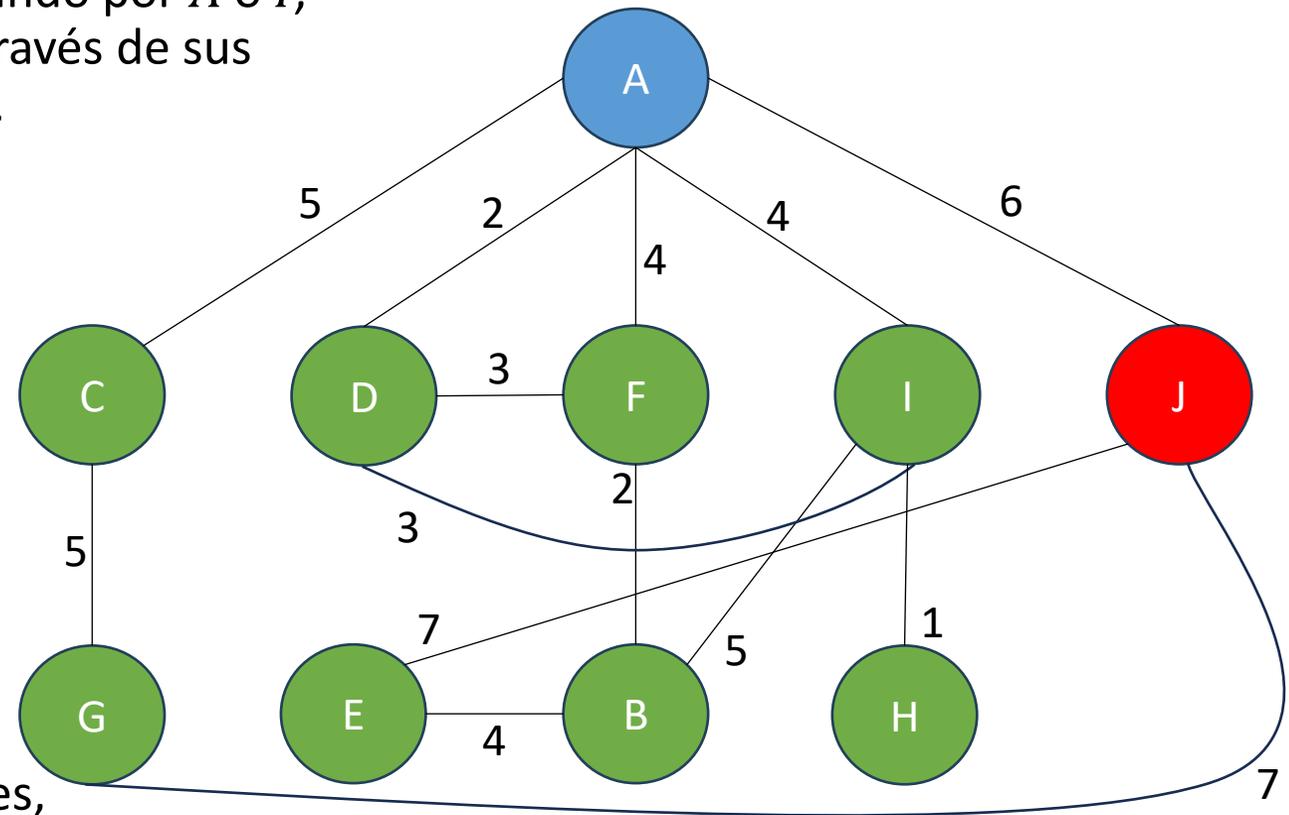
- Para llegar a J desde D , la mejor opción es pasando por A o I , pero la heurística indica que está más cerca a través de sus vecinos con un peso acumulado estimado de 5.
- Heurística $h(D) = 5$.

- **Nodo E :**

- Tiene una conexión directa con J con un peso de 7.
- Heurística $h(E) = 1$ (estimación basada en que J está muy cerca).

- **Nodo F :**

- Desde F , la mejor manera de llegar a J es pasando por B y E , pero F tiene más conexiones, por lo que su heurística es ligeramente mayor.
- Heurística $h(F) = 4$ (basado en los vecinos de menor coste).



Ejercicio 4

- **Nodo G :**

- G está directamente conectado a J con un coste de 7.
- Heurística $h(G) = 3$ (indica que está más cercano al objetivo).

- **Nodo H :**

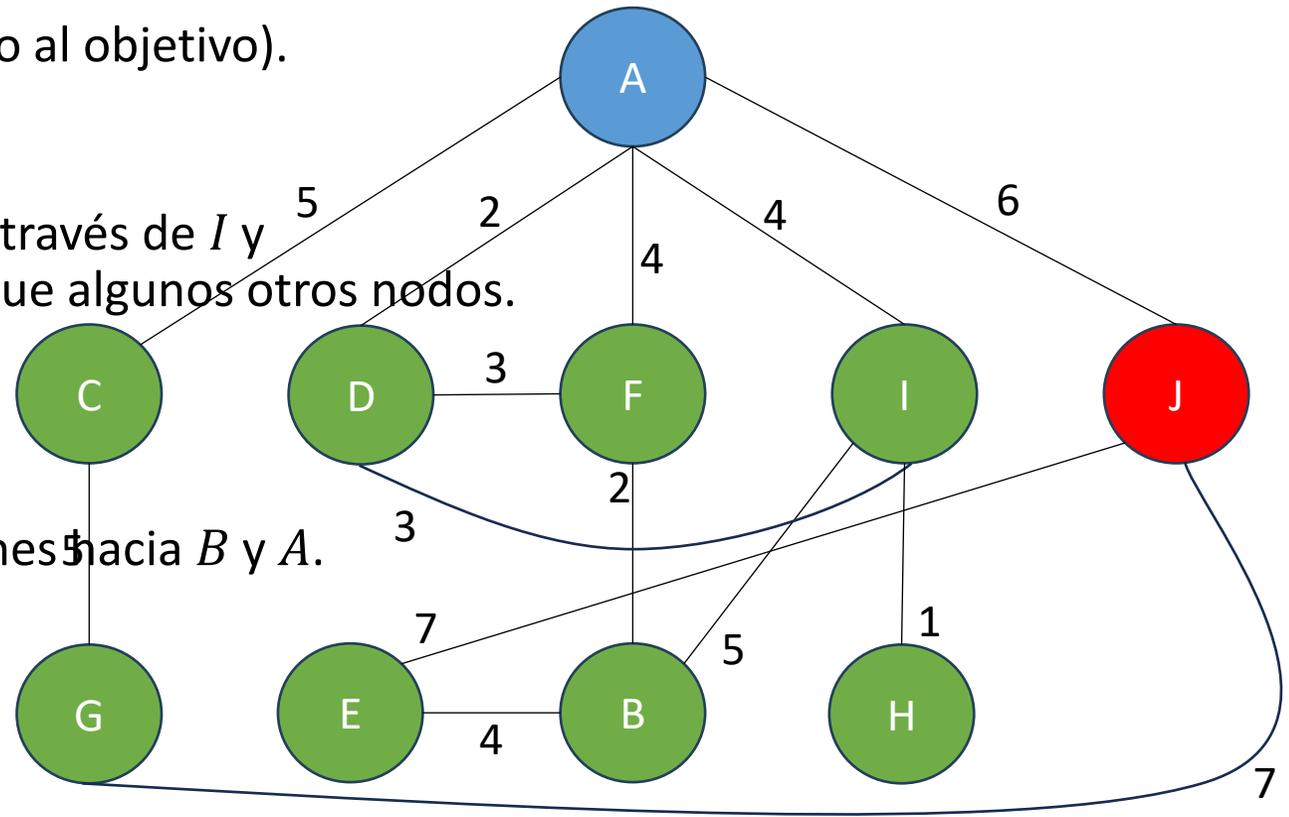
- Para llegar desde H a J , la mejor opción sería a través de I y sus conexiones, pero está estimado más lejos que algunos otros nodos.
- Heurística $h(H) = 2$.

- **Nodo I**

- I está relativamente cerca de J y tiene conexiones hacia B y A .
- Heurística $h(I) = 3$ (estimación basada en su proximidad a través de conexiones eficientes).

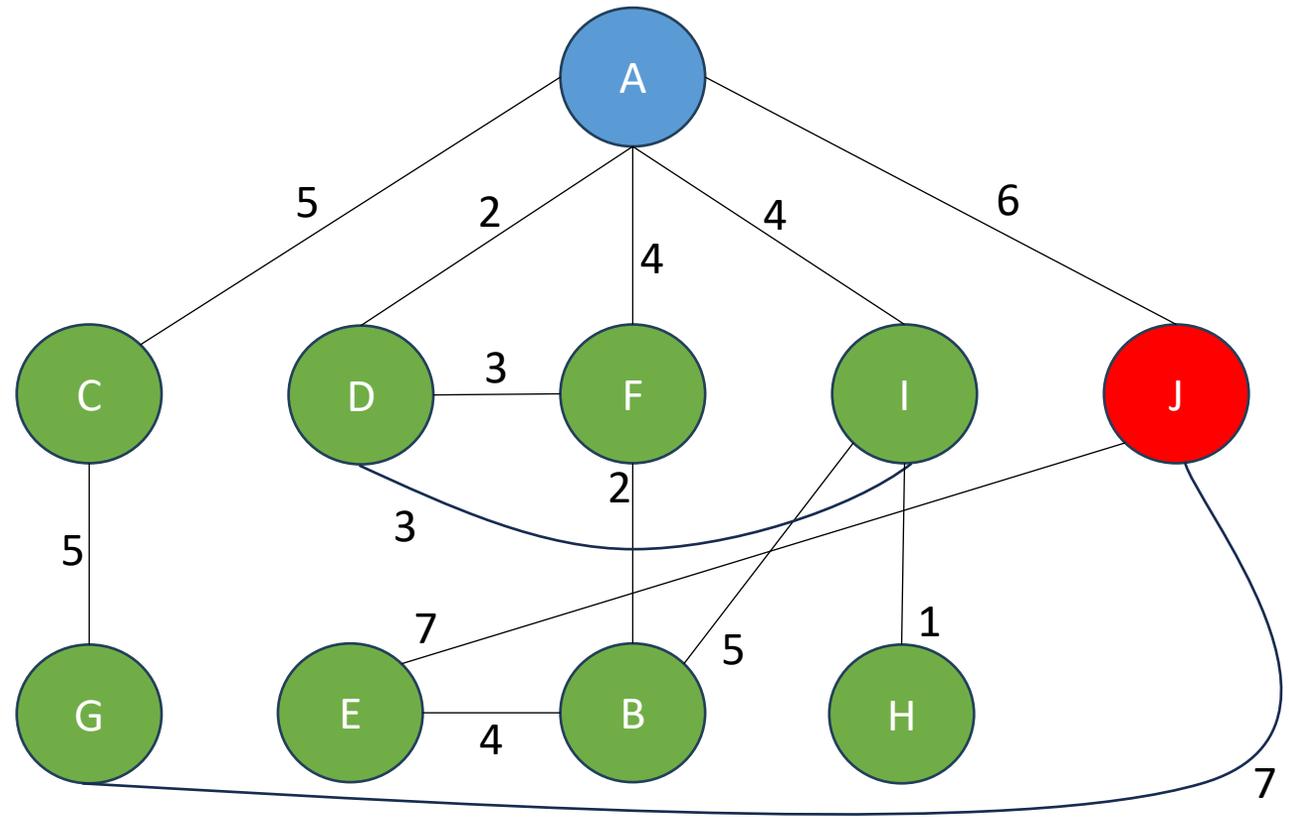
- **Nodo J :**

- Es el nodo objetivo, por lo que su heurística es 0.
- Heurística $h(J) = 0$.



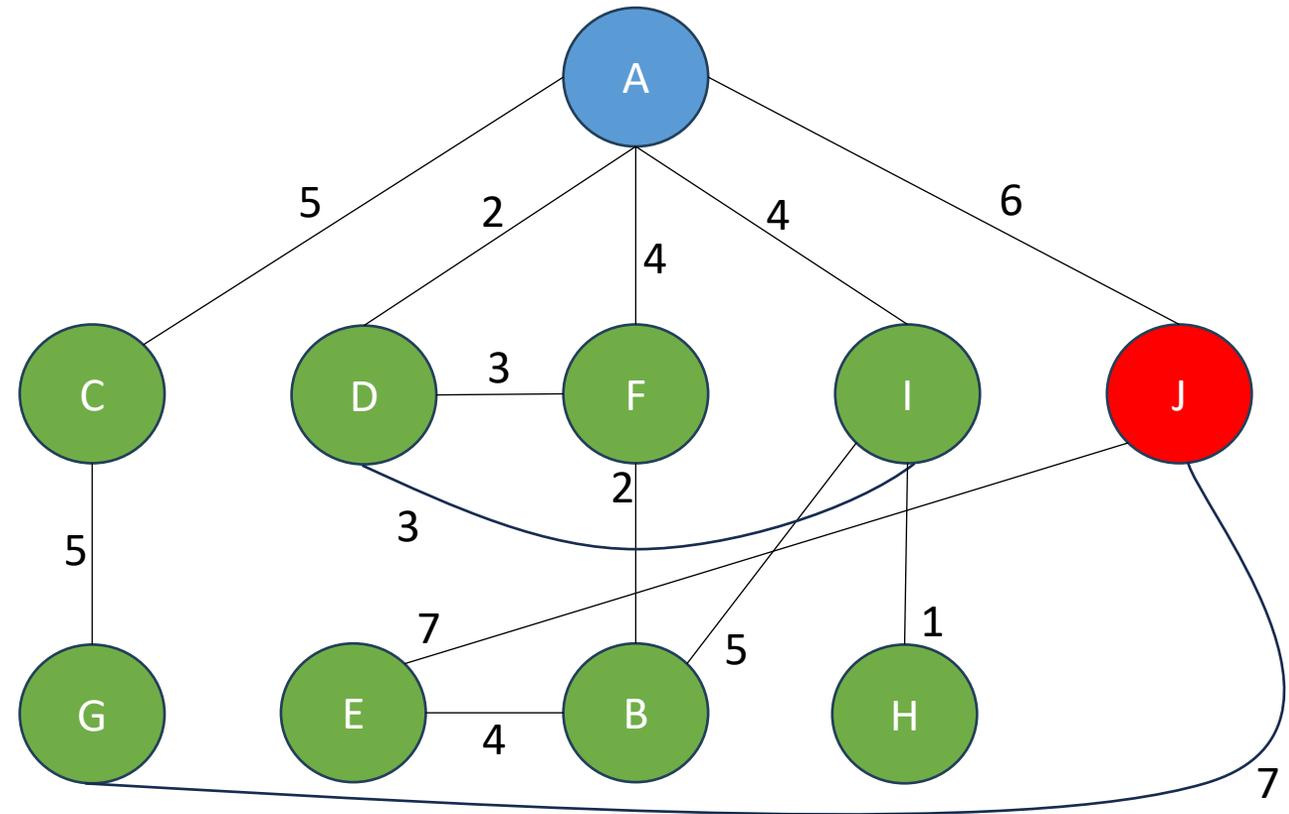
Ejercicio 4

- A : 6 (estimación a J)
- B : 4
- C : 7
- D : 5
- E : 1
- F : 4
- G : 3
- H : 2
- I : 3
- J : 0 (el nodo objetivo)

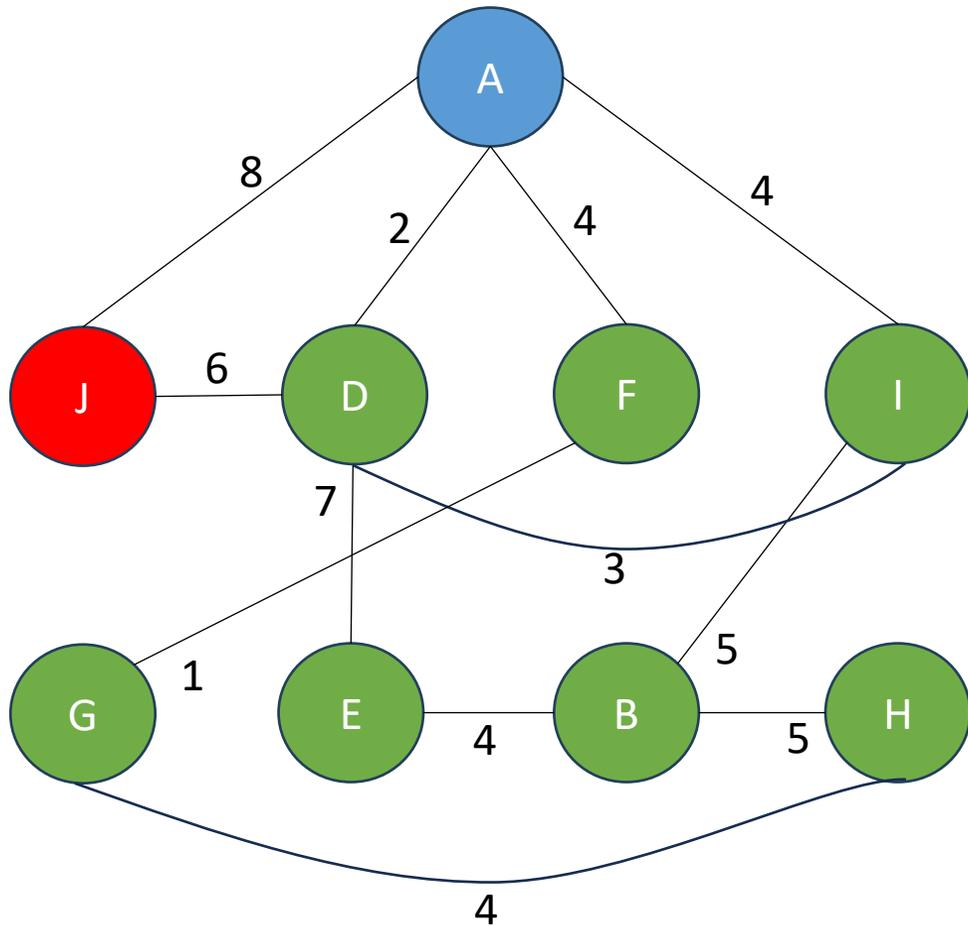


Ejercicio 4

- Desde A :
 - $A \rightarrow C$: $g(C)=5, h(C)=7, f(C)=5+7=12$
 - $A \rightarrow D$: $g(D)=2, h(D)=5, f(D)=2+5=7$
 - $A \rightarrow F$: $g(F)=4, h(F)=4, f(F)=4+4=8$
 - $A \rightarrow I$: $g(I)=4, h(I)=3, f(I)=4+3=7$
 - $A \rightarrow J$: $g(J)=6, h(J)=0, f(J)=6+0=6$
 - Nodo seleccionado: J con $f(J)=6$
(ya que es el nodo con el menor valor de $f(n)$)
- Camino más corto encontrado por A^* : $A \rightarrow J$
- Coste total: 6



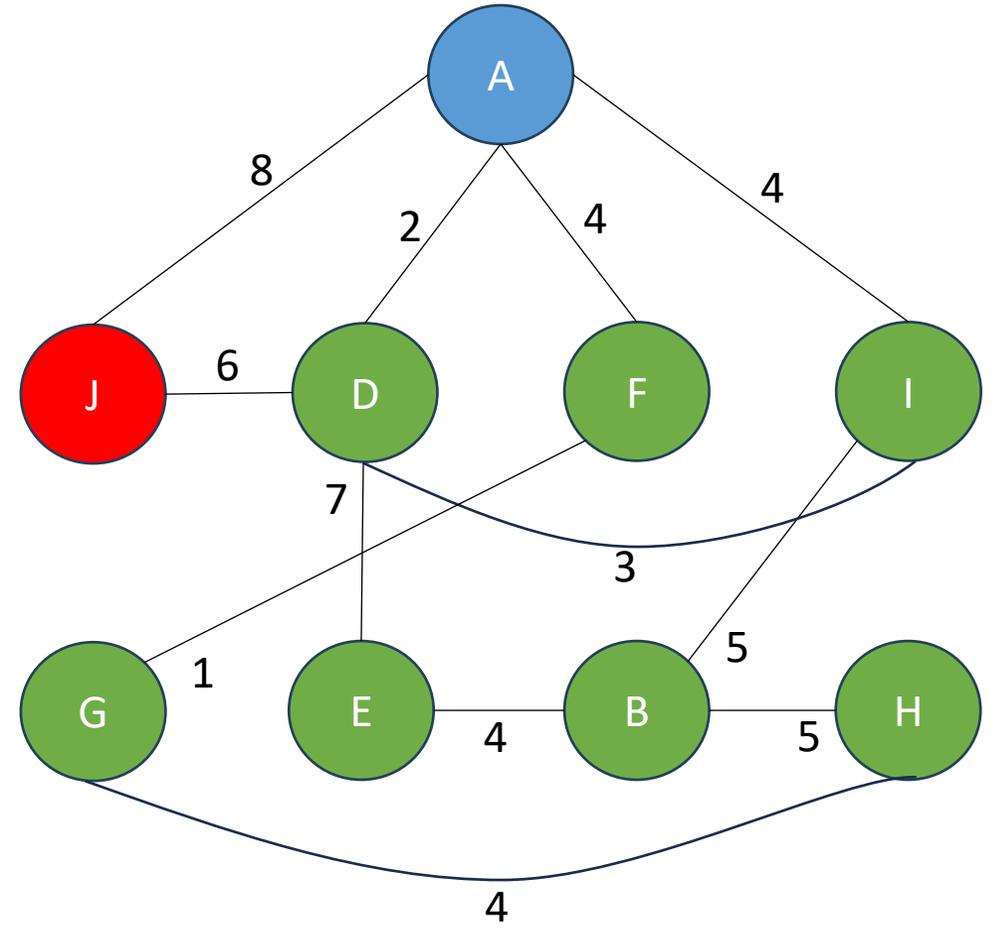
Ejercicio 5



- Sobre el grafo anterior:
 - Siendo A el vértice de inicio.
 - Siendo J el vértice objetivo.
- Se pide aplicar:
 - A*

Ejercicio 5

- A : 1 (directo a J)
- B : 5
- D : 1 (directo a J)
- E : 5
- F : 7
- G : 3
- H : 4
- I : 6
- J : 0 (el nodo objetivo)



Ejercicio 5

- Desde A :

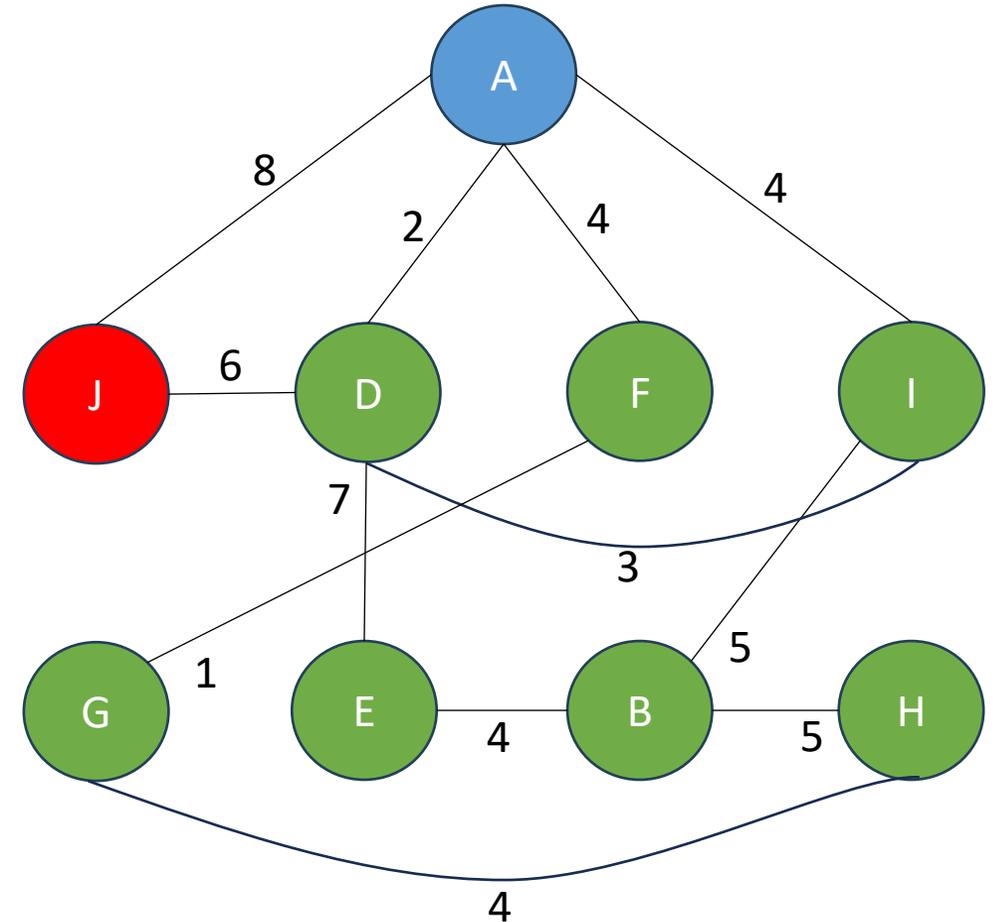
- $A \rightarrow D$: $g(D)=2, h(D)=1, f(D)=2+1=3$
- $A \rightarrow F$: $g(F)=4, h(F)=7, f(F)=4+7=11$
- $A \rightarrow I$: $g(I)=4, h(I)=6, f(I)=4+6=10$
- $A \rightarrow J$: $g(J)=8, h(J)=0, f(J)=8+0=8$
- $A \rightarrow H$: $g(H)=3, h(H)=4, f(H)=3+4=7$
- Nodo seleccionado: D con $f(D)=3$.

- Desde D :

- $D \rightarrow A$ (ya visitado)
- $D \rightarrow I$: $g(I)=g(D)+3=2+3=5, h(I)=6, f(I)=5+6=11$
- $D \rightarrow E$: $g(E)=g(D)+7=2+7=9, h(E)=5, f(E)=9+5=14$
- $D \rightarrow J$: $g(J)=g(D)+6=2+6=8, h(J)=0, f(J)=8+0=8$
- Nodo seleccionado: J con $f(J)=8$.

- Camino final: $A \rightarrow D \rightarrow J$

- Coste total: $2+6=8$



DESARROLLO DE JUEGOS CON INTELIGENCIA ARTIFICIAL

Búsqueda en el espacio de estados y agentes

Tema 4. Búsqueda con heurísticas fuertes

©2023 Autores
David María Arribas,
Sergio Hernández García,
Nicolás H. Rodríguez Uribe,

Algunos derechos reservados
Este documento se distribuye bajo la licencia
"Atribución-CompartirIgual 4.0 Internacional" de Creative Commons, disponible en
<https://creativecommons.org/licenses/by-sa/4.0/deed.es>



Nicolás Rodríguez
nicolas.rodriguez@urjc.es

David María
david.maria@urjc.es

Sergio Hernández
sergio.Hernandez@urjc.es



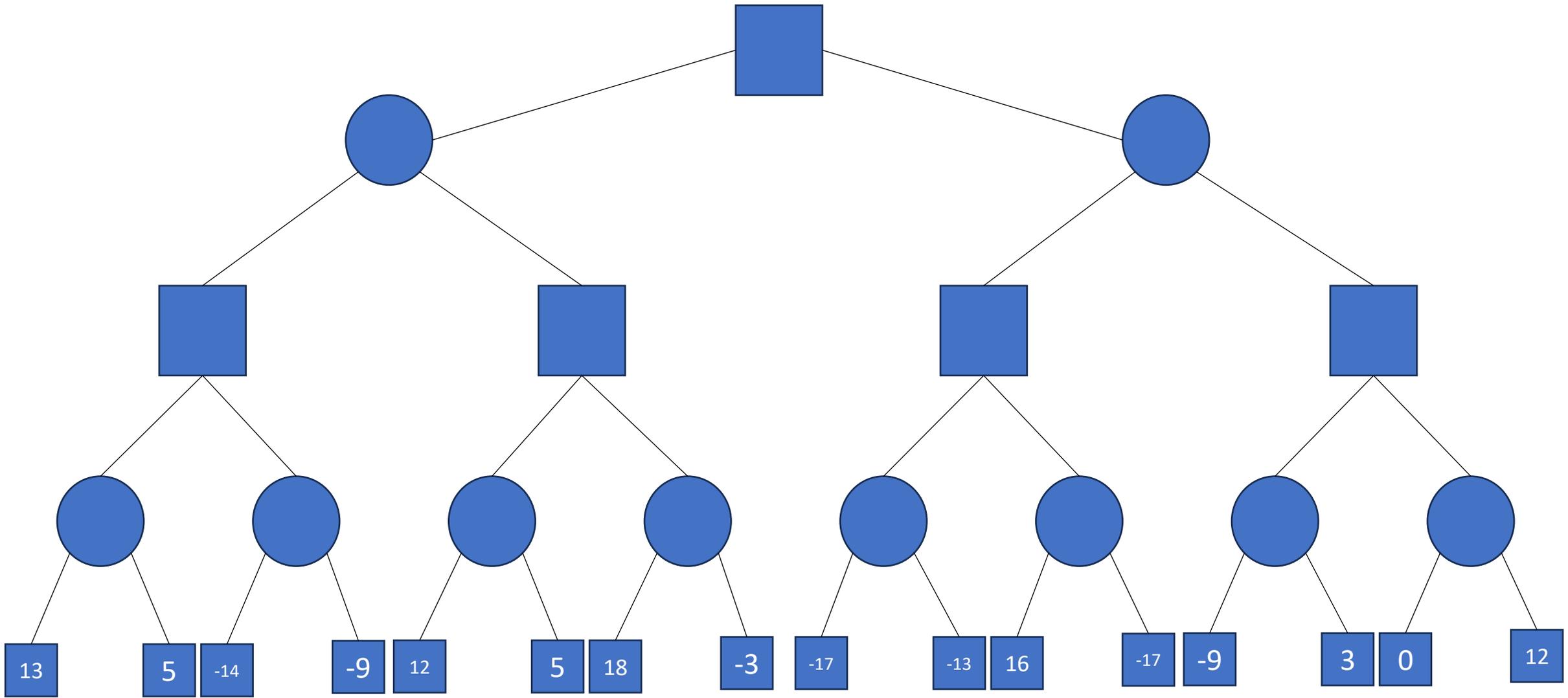
DESARROLLO DE JUEGOS CON INTELIGENCIA ARTIFICIAL

Búsqueda en el espacio de estados y agentes

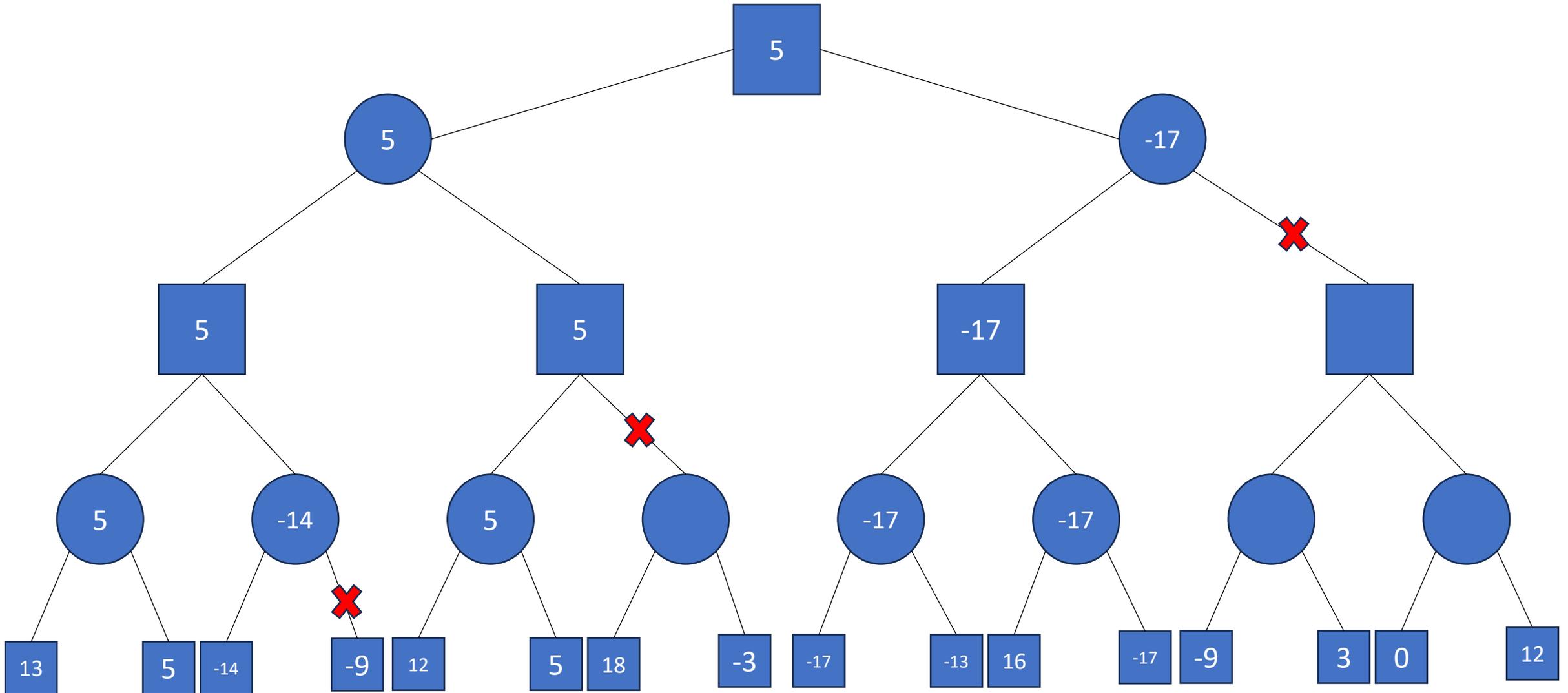
Tema 5. Búsqueda multiagente y suma cero



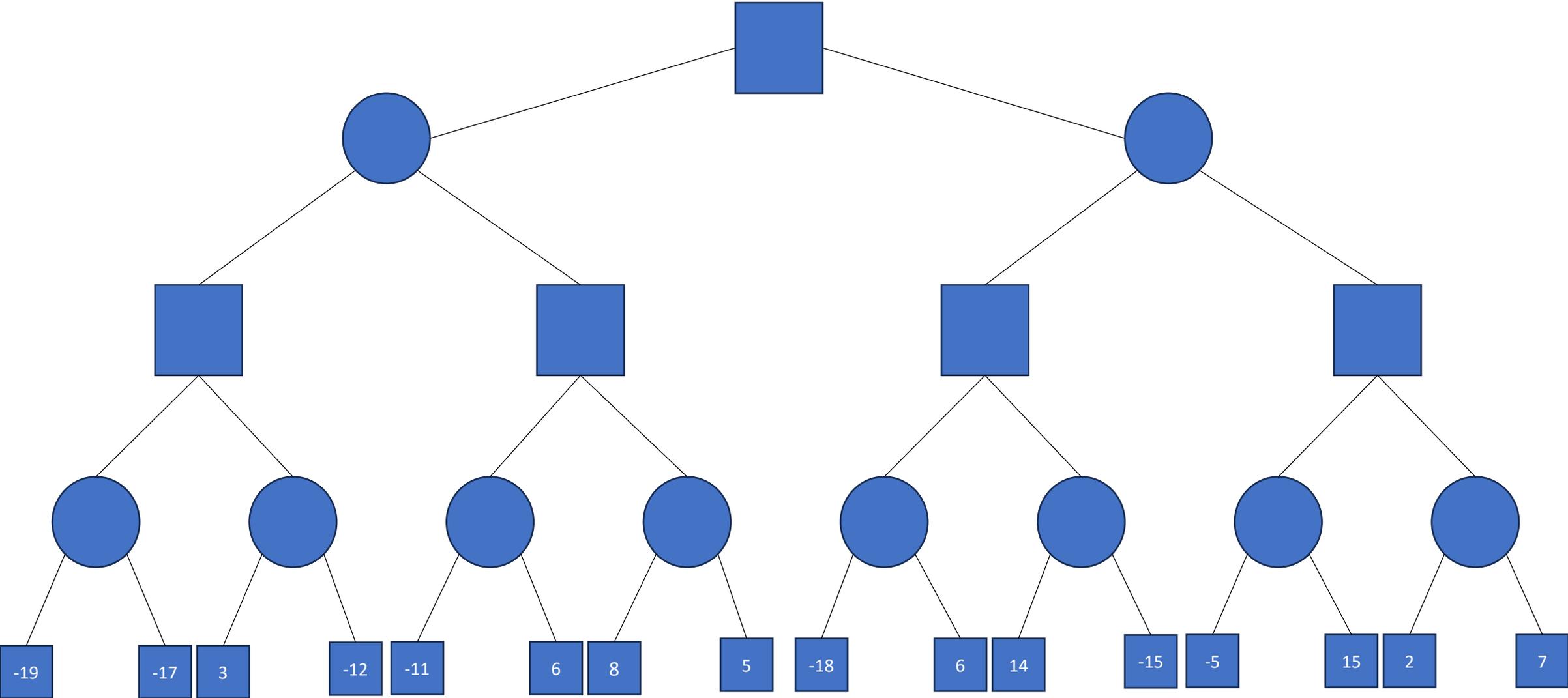
Ejercicio 1



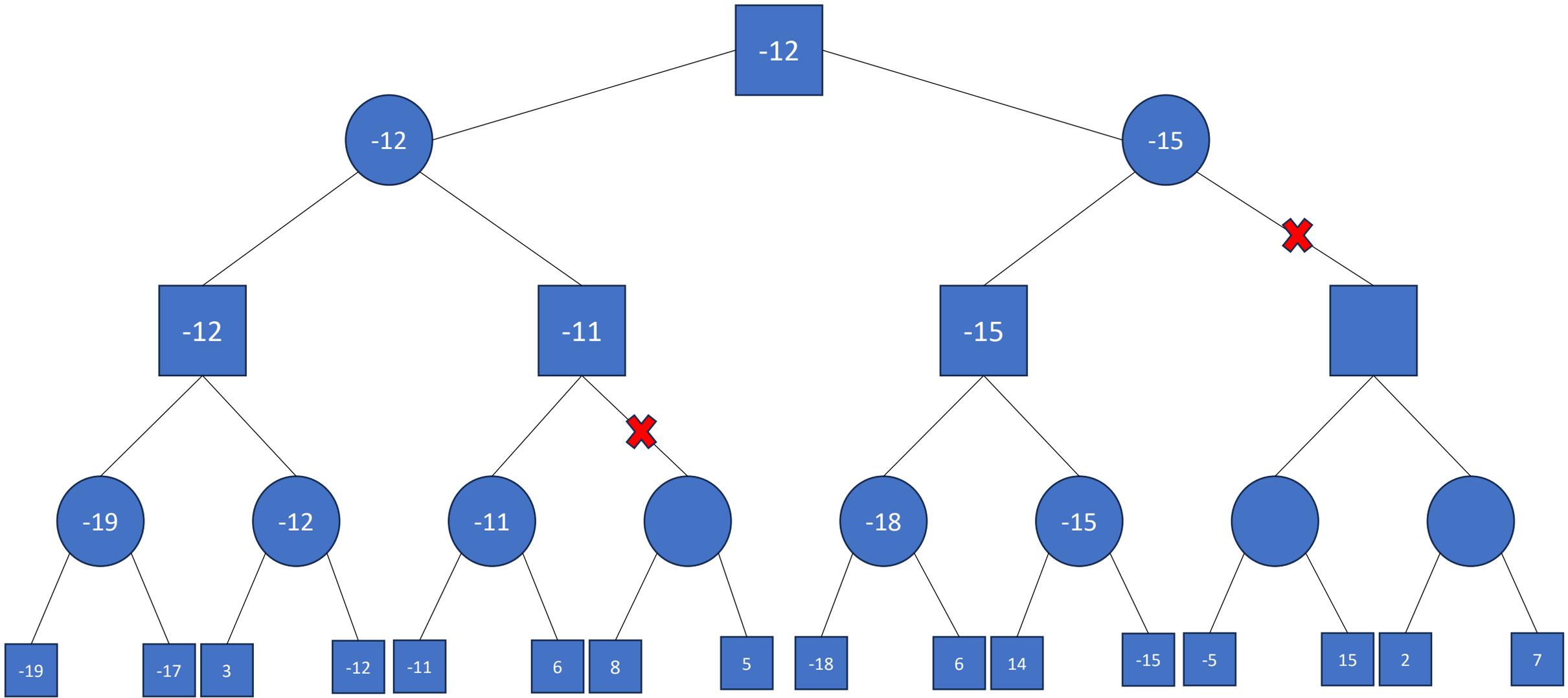
Ejercicio 1



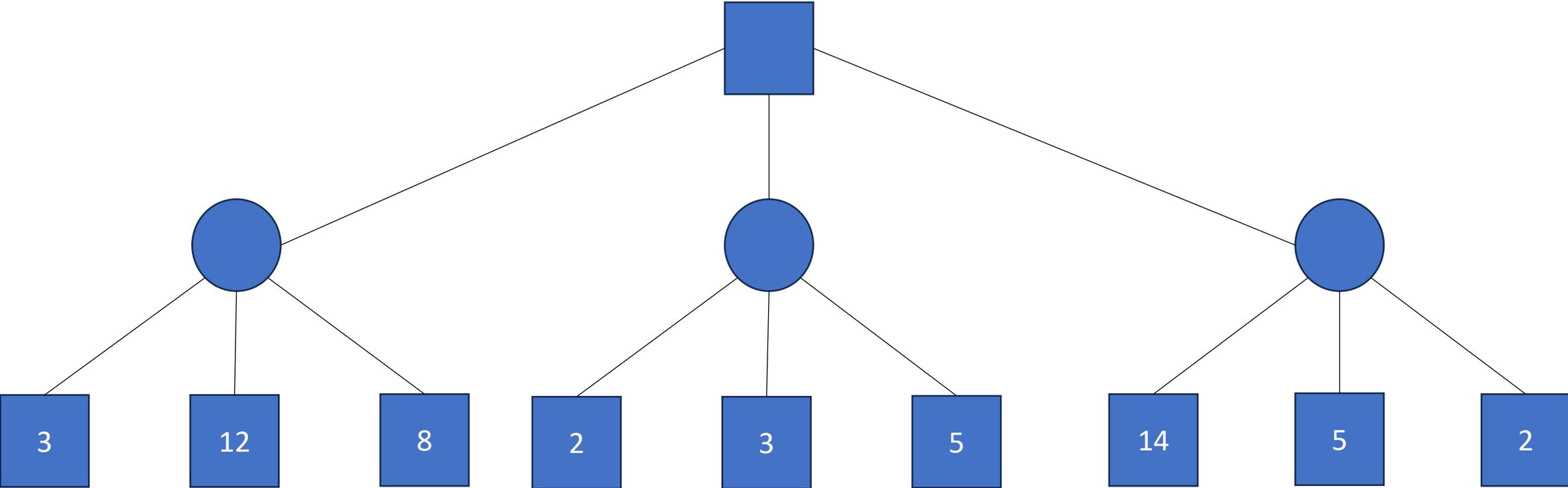
Ejercicio 2



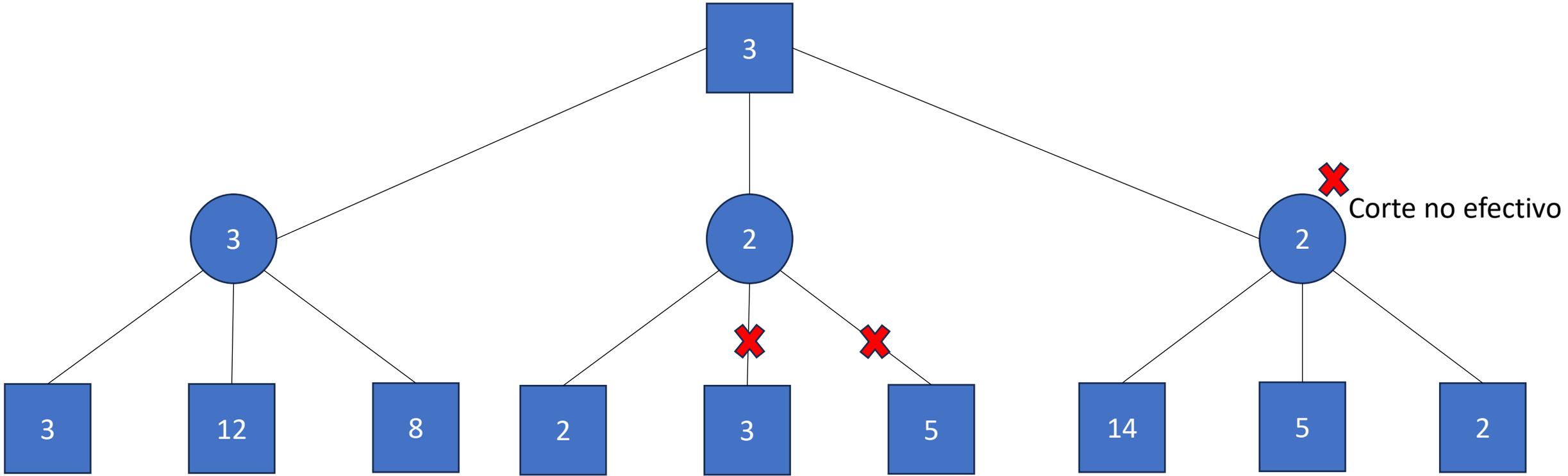
Ejercicio 2



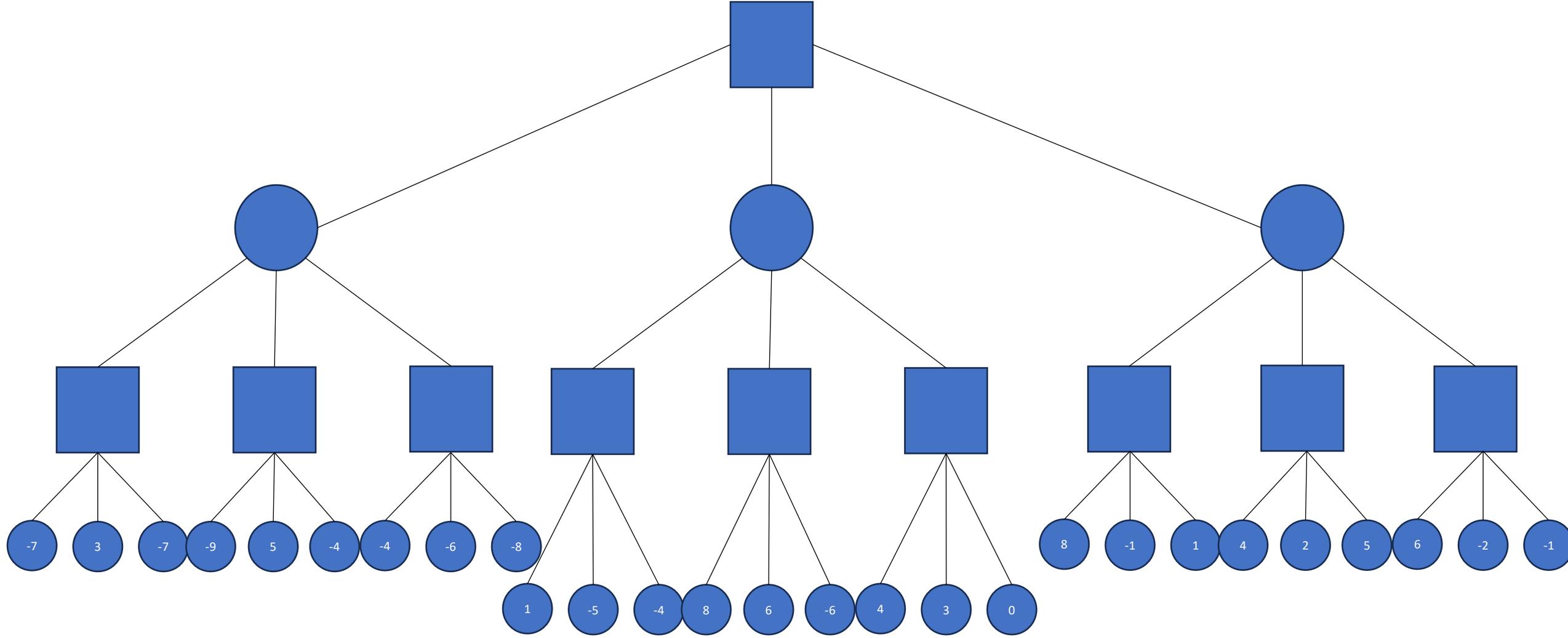
Ejercicio 3



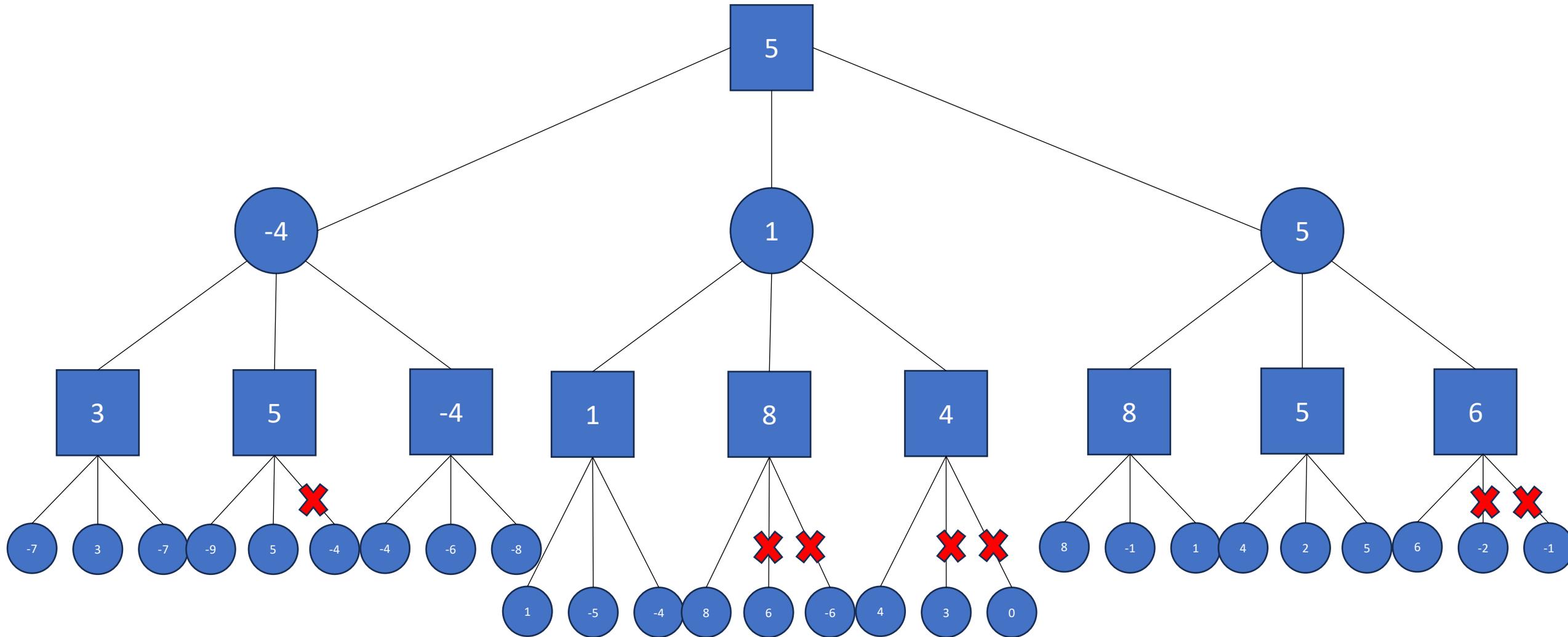
Ejercicio 3



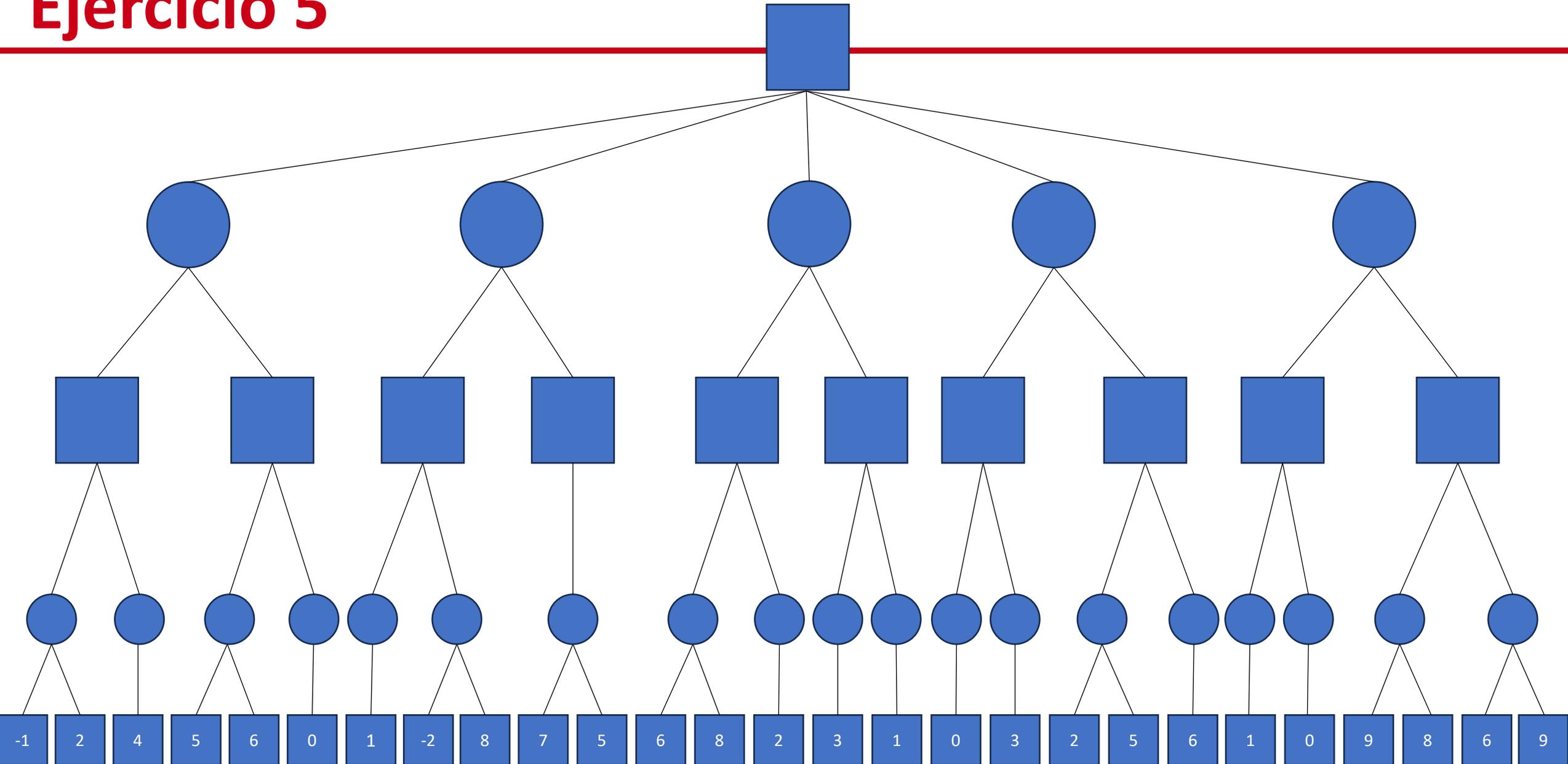
Ejercicio 4



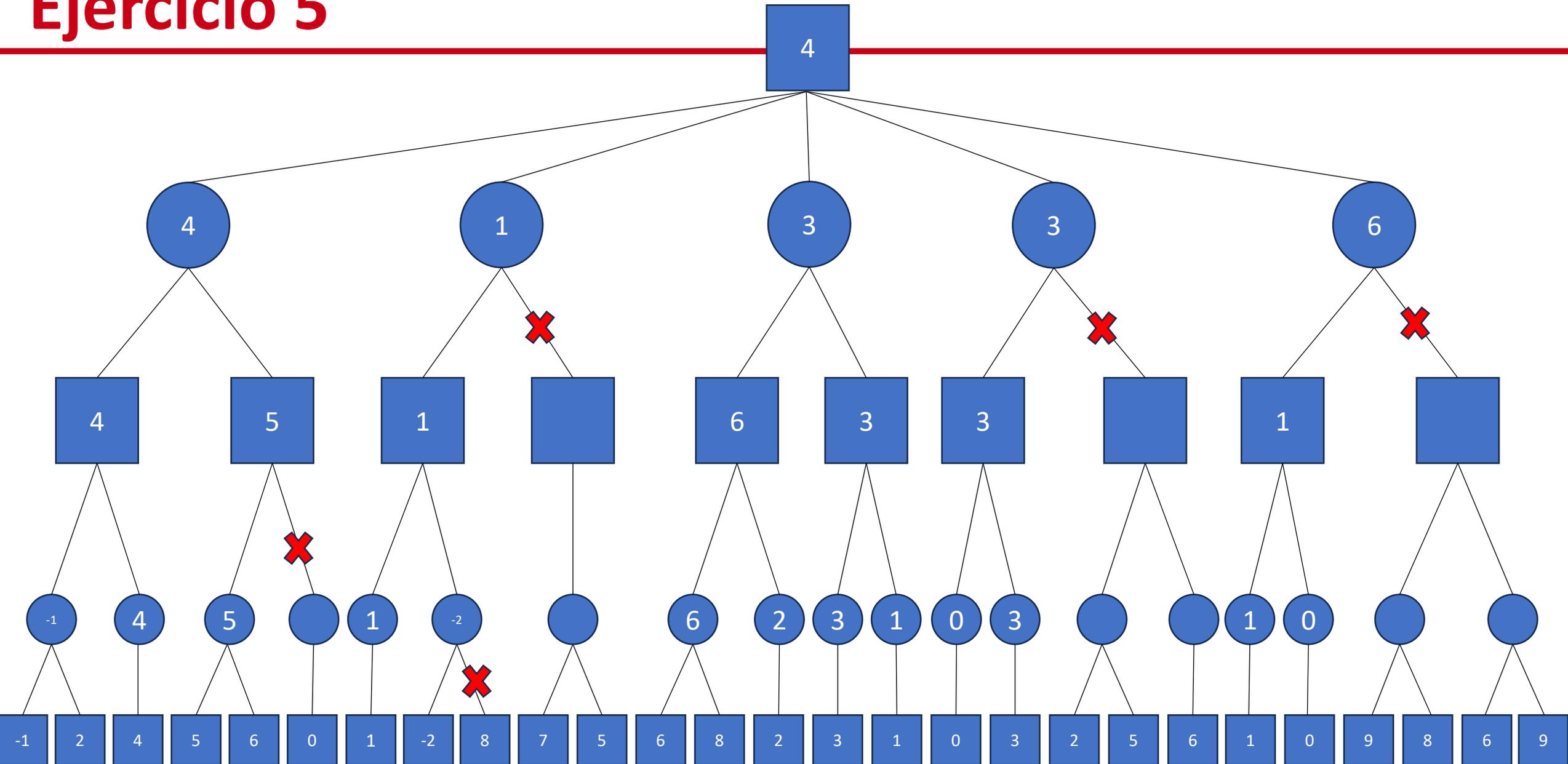
Ejercicio 4



Ejercicio 5



Ejercicio 5



DESARROLLO DE JUEGOS CON INTELIGENCIA ARTIFICIAL

Búsqueda en el espacio de estados y agentes

Tema 5. Búsqueda multiagente y suma cero

©2023 Autores
David María Arribas,
Sergio Hernández García,
Nicolás H. Rodríguez Uribe,

Algunos derechos reservados
Este documento se distribuye bajo la licencia
"Atribución-CompartirIgual 4.0 Internacional" de Creative Commons, disponible en
<https://creativecommons.org/licenses/by-sa/4.0/deed.es>



Nicolás Rodríguez
nicolas.rodriguez@urjc.es

David María
david.maria@urjc.es

Sergio Hernández
sergio.hernandez@urjc.es



DESARROLLO DE JUEGOS CON INTELIGENCIA ARTIFICIAL

Bloque II - Tema 1

Optimización numérica - EJERCICIOS



Ejercicio 1 - Enunciado



$x \setminus y$	0	1	2	3	4	5	6	7	8	9
0	0.1	0.9	0.5	0.6	0.2	0.1	0.3	0.5	0.6	1.2
1	0.1	0.8	1.0	0.6	0.4	0.3	0.2	0.4	0.4	0.8
2	0.3	0.8	2.0	0.7	0.5	0.4	8.6	0.2	0.3	0.3
3	0.4	0.7	2.7	0.9	0.7	0.4	8.3	0.1	0.2	0.1
4	0.7	0.8	3.6	1.9	0.9	0.8	7.9	7.5	7.2	0.1
5	0.8	1.1	3.9	1.8	1.2	1.1	0.9	1.2	6.9	0.2
6	0.8	1.2	4.0	4.2	3.2	5.1	5.3	6.3	6.1	0.2
7	0.9	1.2	2.0	4.4	4.6	4.7	3.1	2.4	2.9	1.0
8	1.0	1.3	1.2	1.0	0.8	0.7	0.4	0.3	1.3	1.0
9	1.2	1.1	0.9	0.9	0.7	0.6	0.6	0.4	0.4	1.3

En la imagen de arriba el grupo de arqueros enmarcado en rojo pertenece a nuestro enemigo, y lo gobierna una inteligencia artificial (IA). Suponer que la IA ha agrupado a estos arqueros de modo que todos disparan a la vez sobre su blanco. Dicho blanco se selecciona como la celda del terreno donde mayor daño nos hace el ataque. Los valores de fitness de las diferentes celdas del terreno se encuentran en la tabla a su derecha.

Ejercicio 1 – Pregunta 1

Encontrar una posición de disparo utilizando *Hill Climbing*, arrancando en la posición (1,1).

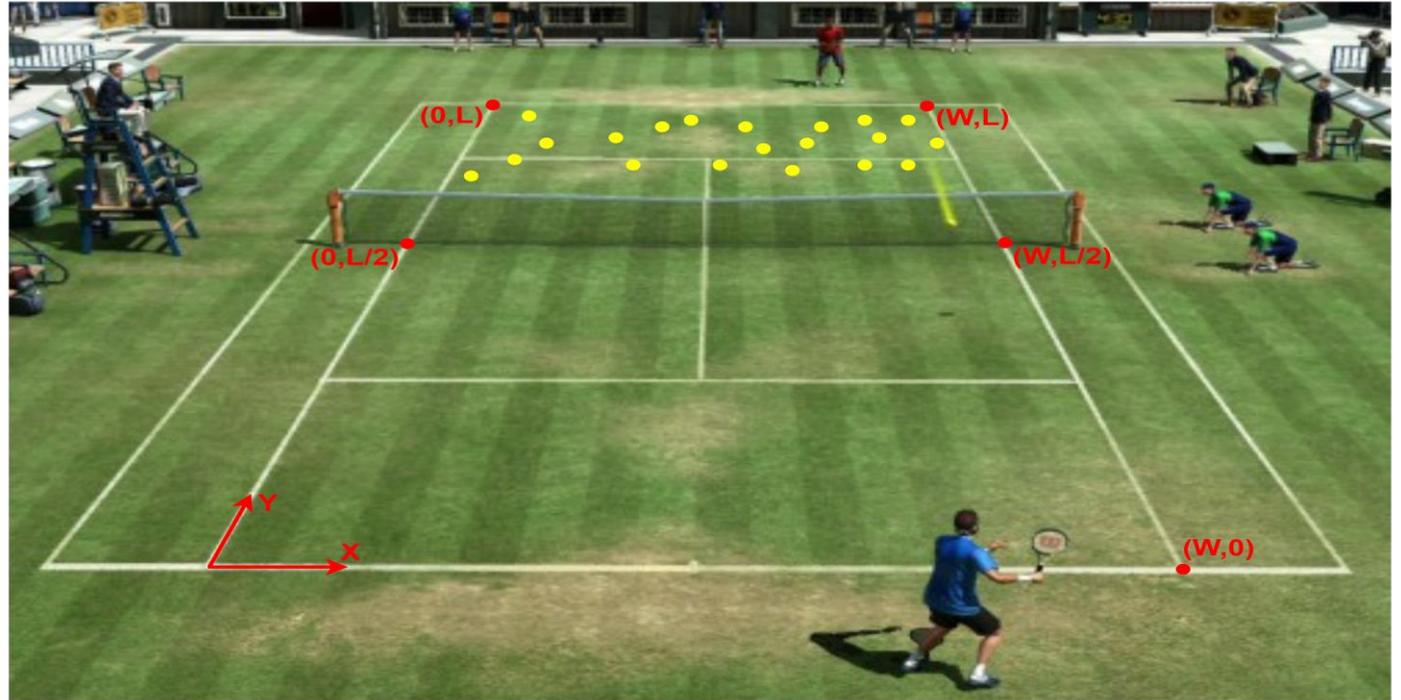
POSICIÓN	DIFF. FITNESS	DIRECCIÓN
(1,1)	0.2	E
(1,2)	1.0	S
(2,2)	0.7	S
(3,2)	0.9	S
(4,2)	0.3	S
(5,2)	0.1	S
(6,2)	0.2	E
(6,3)	0.2	S
(7,3)	0.2	E
(7,4)	0.1	E
(7,5)	0.4	N
(6,5)	0.2	E
(6,6)	1.0	E
(6,7)	--	--

x\y	0	1	2	3	4	5	6	7	8	9
0	0.1	0.9	0.5	0.6	0.2	0.1	0.3	0.5	0.6	1.2
1	0.1	0.8	1.0	0.6	0.4	0.3	0.2	0.4	0.4	0.8
2	0.3	0.8	2.0	0.7	0.5	0.4	8.6	0.2	0.3	0.3
3	0.4	0.7	2.7	0.9	0.7	0.4	8.3	0.1	0.2	0.1
4	0.7	0.8	3.6	1.9	0.9	0.8	7.9	7.5	7.2	0.1
5	0.8	1.1	3.9	1.8	1.2	1.1	0.9	1.2	6.9	0.2
6	0.8	1.2	4.0	4.2	3.2	5.1	5.3	6.3	6.1	0.2
7	0.9	1.2	2.0	4.4	4.6	4.7	3.1	2.4	2.9	1.0
8	1.0	1.3	1.2	1.0	0.8	0.7	0.4	0.3	1.3	1.0
9	1.2	1.1	0.9	0.9	0.7	0.6	0.6	0.4	0.4	1.3

Ejercicio 2 - Enunciado

Calculadora de fitness

Dims. de medio campo		candidato	
W	L/2	x0	y0
8.3	11.9	8.1	11.9
Dimensiones 1/2 campo		Candidato	
W	L/2	x0	y0
8.3	11.9	8.1	11.9
Coordenadas de los impactos de pelota		Fitness	
x	y	0.024070	
F(x0,y0)			
2	12	6.85E-17	
3	14	6.15E-14	
4	19	6.41E-30	
3	18	3.50E-28	
5	14	8.15E-07	
6	12	1.20E-02	
2	20	2.22E-45	
5	21	7.29E-41	
3	17	2.56E-23	
1	17	6.47E-34	
2	12	6.85E-17	
3	14	6.15E-14	
4	19	6.41E-30	
3	18	3.50E-28	
5	14	8.15E-07	
6	12	1.20E-02	



En un juego de tenis, la posición de nuestro avatar (el jugador azul) se mide desde la esquina inferior izquierda de nuestro campo. Por ejemplo, en la imagen de abajo el avatar está, aproximadamente, en la posición $(x_0, y_0) = (5, -0,8)$. La IA se encarga, entre otras cosas, de mover al contrario a la mejor posición (x^*, y^*) para recibir la pelota que le devolvamos. El fitness de la posición es la probabilidad de ganar en esa posición, dónde enviaremos la pelota y se calcula teniendo en cuenta nuestras últimas 20 devoluciones.

Ejercicio 2 - Pregunta 1

Utilizar *Hill Climbing* para seleccionar la posición donde desplazar al contrario, con un paso base de 0.5 tanto en el eje X como en el eje Y; y arrancando en la posición (W, L/2)

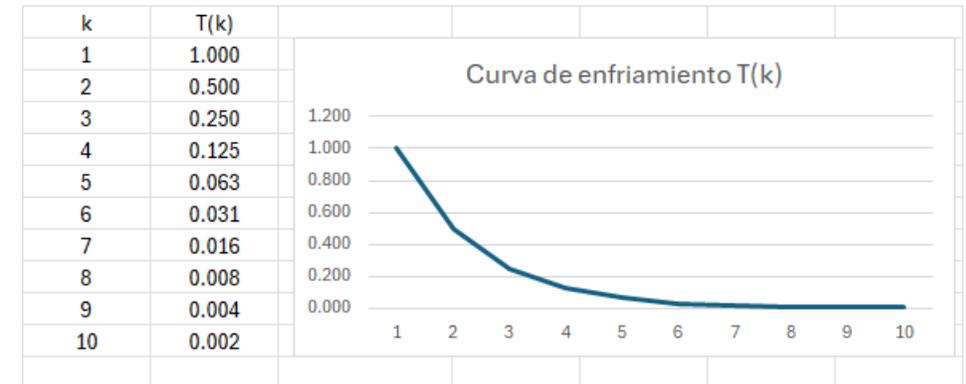
POSICIÓN	DIFF. FITNESS	DIRECCION
(8.30, 11.90)	0.070	W
(7.80, 11.90)	0.290	W
(7.30, 11.90)	0.680	W
(6.80, 11.90)	0.770	W
(6.30, 11.90)	0.100	W
(5.80, 11.90)	--	--

Ejercicio 2 - Pregunta 2

Emplear ahora ***Hill Climbing con paso variable*** avanzando un 35% extra en cada iteración y con un paso base de 0.25 tanto en el eje X como en el eje Y; y arrancando en la posición (W, L/2)

Ejercicio 2 - Pregunta 3

Utilizar *Hill Climbing con Templado probabilístico* con una curva de enfriamiento arrancando en la misma posición que en los ejercicios anteriores, con paso de 0.5 y la curva de enfriamiento es $T(k) = T(k-1)/2$; con $T(1)=1$. Ejecuta 10 pasos en total.



DESARROLLO DE JUEGOS CON INTELIGENCIA ARTIFICIAL

Bloque II - Tema 1

Optimización numérica - EJERCICIOS



©2023 Autores
David María Arribas,
Sergio Hernández García,
Nicolás H. Rodríguez Uribe,
Algunos derechos reservados
Este documento se distribuye bajo la licencia
"Atribución-CompartirIgual 4.0 Internacional" de Creative Commons,
disponible en
<https://creativecommons.org/licenses/by-sa/4.0/deed.es>

David María
david.maria@urjc.es

Sergio Hernández
sergio.hernandez@urjc.es

Nicolás Rodríguez
nicolas.rodriguez@urjc.es



DESARROLLO DE JUEGOS CON INTELIGENCIA ARTIFICIAL

Bloque II - Tema 2

Redes Neuronales – EJERCICIOS Soluciones



David María
david.maria@urjc.es

Sergio Hernández
sergio.hernandez@urjc.es

Nicolás Rodríguez
nicolas.rodriguez@urjc.es



Ejercicio 1: selección de acciones

En un juego de Baloncesto queremos construir la IA del equipo blanco (Boston Celtics) con redes neuronales. El objetivo es modelar la actividad del NPC que lleva la pelota para que elija la acción a realizar cada 5 segundos. Las acciones que puede realizar son: **pasar al jugador 1, pasar al jugador 2, pasar al jugador 3, pasar al jugador 4, avanzar a canasta y lanzar a canasta desde la posición.**

Para ello dispone de un conjunto de datos que recopila una serie de escenarios asociados a cada una de estas acciones. Cada escenario consiste en: **un atributo booleano por cada uno de los jugadores de Boston** que indica si está marcado por un jugador del equipo azul (LA Lakers), **un atributo booleano** que indica si Boston va por delante y **un atributo booleano** que toma 1 si el tiempo de posesión que aún queda es superior a 12 segundos.



Pregunta 1.1

- ¿Cuántas neuronas de entrada y salida son necesarias?
- ¿Qué tamaño tendrá la red neuronal con 1 capa oculta con 5 neuronas?

7 neuronas de entrada, una por cada atributo booleano y 6 neuronas de salida, una por cada posible acción.

Pregunta 1.2

- Según las características de la pregunta anterior, y suponiendo que todos los pesos se inicializan a 0. **Calcula los nuevos pesos actualizados tras introducir el primer ejemplo:**

	M1	M2	M3	M4	M5	BG	TP	P1	P2	P3	P4	A	L
Ej. 1	0	0	1	0	1	0	1	0	1	0	0	1	0

Pregunta 1.3

- Calcula de nuevo los pesos actualizados si introducimos el segundo ejemplo, que es:

	M1	M2	M3	M4	M5	BG	TP	P1	P2	P3	P4	A	L
Ej. 2	1	0	0	1	0	0	0	1	0	1	0	0	1

Solución pregunta 1.2 y 1.3

eta 0.3													
	M1	M2	M3	M4	M5	BG	TP	P1	P2	P3	P4	A	L
Ej.1	0	0	1	0	1	0	1	0	1	0	0	1	0
Ej.2	1	0	0	1	0	0	0	1	0	1	0	0	1

BIAS 1												
Entradas												
	M1	M2	M3	M4	M5	BG	TP	BIAS	SUMA	SIGM.	delta	
Ej.1	0	0	1	0	1	0	1	0	0.000	0.50000	0.004	
H1	0	0	0	0	0	0	0	0	0.000	0.50000	0.004	
H2	0	0	0	0	0	0	0	0	0.000	0.50000	0.004	
H3	0	0	0	0	0	0	0	0	0.000	0.50000	0.004	
H4	0	0	0	0	0	0	0	0	0.000	0.50000	0.004	
H5	0	0	0	0	0	0	0	0	0.000	0.50000	0.004	

BIAS 1												
Entradas												
	M1	M2	M3	M4	M5	BG	TP	BIAS	SUMA	SIGM.	delta	
Ej.2	1	0	0	1	0	0	0	1	0.001	0.50026	0.0012	
H1	0	0	0.00105	0	0.00105	0	0.00105	0.00105	0.001	0.50026	0.0012	
H2	0	0	0.00105	0	0.00105	0	0.00105	0.00105	0.001	0.50026	0.0012	
H3	0	0	0.00105	0	0.00105	0	0.00105	0.00105	0.001	0.50026	0.0012	
H4	0	0	0.00105	0	0.00105	0	0.00105	0.00105	0.001	0.50026	0.0012	
H5	0	0	0.00105	0	0.00105	0	0.00105	0.00105	0.001	0.50026	0.0012	

BIAS 1												
Entradas												
	M1	M2	M3	M4	M5	BG	TP	BIAS	SUMA	SIGM.	delta	
Ej.1	0.50000	0.50000	0.50000	0.50000	0.50000			1	0.000	0.50000	0	
O1	0	0	0	0	0	0	0	0	0.000	0.50000	0	-0.125
O2	0	0	0	0	0	0	0	0	0.000	0.50000	1	0.125
O3	0	0	0	0	0	0	0	0	0.000	0.50000	0	-0.125
O4	0	0	0	0	0	0	0	0	0.000	0.50000	0	-0.125
O5	0	0	0	0	0	0	0	0	0.000	0.50000	1	0.125
O6	0	0	0	0	0	0	0	0	0.000	0.50000	0	-0.125

BIAS 1												
Entradas												
	M1	M2	M3	M4	M5	BG	TP	BIAS	SUMA	SIGM.	delta	
Ej.2	0.50026	0.50026	0.50026	0.50026	0.50026			1	-0.084	0.47891	1	0.1300
O1	-0.01875	-0.01875	-0.01875	-0.01875	-0.01875	-0.03750	-0.03750	-0.03750	-0.084	0.47891	1	0.1300
O2	0.01875	0.01875	0.01875	0.01875	0.01875	0.03750	0.03750	0.03750	0.084	0.52108	0	-0.1300
O3	-0.01875	-0.01875	-0.01875	-0.01875	-0.01875	-0.03750	-0.03750	-0.03750	-0.084	0.47892	1	0.1300
O4	-0.01875	-0.01875	-0.01875	-0.01875	-0.01875	-0.03750	-0.03750	-0.03750	-0.084	0.47892	0	-0.1195
O5	0.01875	0.01875	0.01875	0.01875	0.01875	0.03750	0.03750	0.03750	0.084	0.52108	0	-0.1300
O6	-0.01875	-0.01875	-0.01875	-0.01875	-0.01875	-0.03750	-0.03750	-0.03750	-0.084	0.47892	1	0.1300

BIAS 1												
Entradas												
	M1	M2	M3	M4	M5	BG	TP	BIAS	SUMA	SIGM.	delta	
Ej.1	0.00037	0	0.00105	0.00037	0.00105	0	0.00105	0.00142				
H1	0.00037	0	0.00105	0.00037	0.00105	0	0.00105	0.00142				
H2	0.00037	0	0.00105	0.00037	0.00105	0	0.00105	0.00142				
H3	0.00037	0	0.00105	0.00037	0.00105	0	0.00105	0.00142				
H4	0.00037	0	0.00105	0.00037	0.00105	0	0.00105	0.00142				
H5	0.00037	0	0.00105	0.00037	0.00105	0	0.00105	0.00142				

BIAS 1												
Entradas												
	M1	M2	M3	M4	M5	BG	TP	BIAS	SUMA	SIGM.	delta	
Ej.2	0.00077	0.00077	0.00077	0.00077	0.00077	0.00151	0.00151	0.00151				
O1	0.00077	0.00077	0.00077	0.00077	0.00077	0.00151	0.00151	0.00151				
O2	-0.00077	-0.00077	-0.00077	-0.00077	-0.00077	-0.00151	-0.00151	-0.00151				
O3	0.00077	0.00077	0.00077	0.00077	0.00077	0.00151	0.00151	0.00151				
O4	-0.03669	-0.03669	-0.03669	-0.03669	-0.03669	-0.07336	-0.07336	-0.07336				
O5	-0.00077	-0.00077	-0.00077	-0.00077	-0.00077	-0.00151	-0.00151	-0.00151				
O6	0.00077	0.00077	0.00077	0.00077	0.00077	0.00151	0.00151	0.00151				

Pregunta 1.4

- ¿Qué decisión tomará el jugador controlado por la IA ante la siguiente situación?

	M1	M2	M3	M4	M5	BG	TP
Ej. Nuevo	1	1	0	1	0	1	1

- Para ello, asume que la red neuronal ya entrenada tiene los siguientes pesos:

In/Out	H1	H2	H3	H4	H5
M1	0.706	0.641	0.362	0.843	0.565
M2	0.488	0.403	0.942	0.151	0.082
M3	0.883	0.474	0.501	0.617	0.997
M4	0.059	0.227	0.526	0.686	0.118
M5	0.055	0.870	0.830	0.774	0.566
BG	0.367	0.850	0.819	0.373	0.091
TP	0.445	0.602	0.714	0.989	0.008

In/Out	P1	P2	P3	P4	A	L
H1	0.462	0.401	0.113	0.582	0.176	0.279
H2	0.736	0.179	0.342	0.760	0.145	0.996
H3	0.582	0.860	0.554	0.280	0.630	0.200
H4	0.028	0.103	0.717	0.541	0.457	0.849
H5	0.718	0.096	0.554	0.757	0.238	0.063

Pregunta 1.4

- ¿Qué decisión tomará el jugador controlado por la IA ante la siguiente situación?

	M1	M2	M3	M4	M5	BG	TP
Ej. Nuevo	1	1	0	1	0	1	1

- Para ello, asume que la red neuronal ya entrenada tiene los siguientes pesos:

In/Out	H1	H2	H3	H4	H5
M1	0.706	0.641	0.362	0.843	0.565
M2	0.488	0.403	0.942	0.151	0.082
M3	0.883	0.474	0.501	0.617	0.997
M4	0.059	0.227	0.526	0.686	0.118
M5	0.055	0.870	0.830	0.774	0.566
BG	0.367	0.850	0.819	0.373	0.091
TP	0.445	0.602	0.714	0.989	0.008

In/Out	P1	P2	P3	P4	A	L
H1	0.462	0.401	0.113	0.582	0.176	0.279
H2	0.736	0.179	0.342	0.760	0.145	0.996
H3	0.582	0.860	0.554	0.280	0.630	0.200
H4	0.028	0.103	0.717	0.541	0.457	0.849
H5	0.718	0.096	0.554	0.757	0.238	0.063

Suma ponderada	2.194	1.521	2.031	2.549	1.505	2.230
Salida sigmoide	0.900	0.821	0.884	0.928	0.818	0.903

Suma ponderada	2.065	2.724	3.363	3.043	0.865
Salida sigmoide	0.887	0.938	0.967	0.954	0.704

La salida con mayor valor tras aplicar la sigmoide es P4 y por lo tanto esta será la acción que tomará el agente

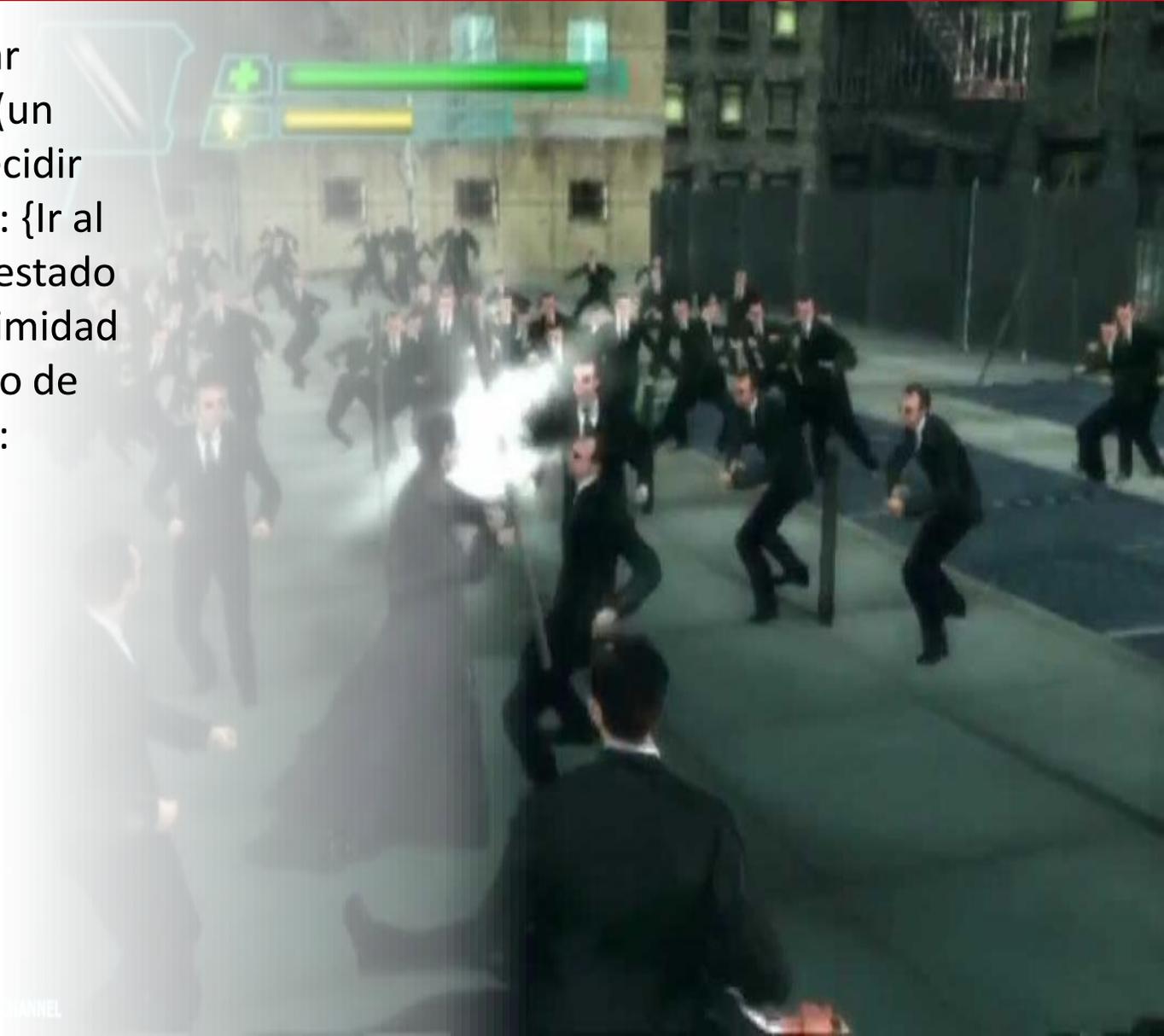
Ejercicio 2: definir una red neuronal

En un juego basado en Matrix, donde nuestro avatar coopera con Trinity (otro jugador en red) y Morfeo (un NPC), queremos entrenar una red neuronal para decidir qué acción debe realizar Morfeo entre dos posibles: {Ir al oráculo, Buscar teléfono}. La decisión depende del estado de nuestro avatar, del estado de Trinity y de la proximidad del agente Smith. Para ello disponemos del conjunto de entrenamiento que se muestra en la tabla de abajo:

Nuestro avatar	Trinity	Smith cerca	Morfeo
Fuera	Capturada	No	Ir al oráculo
Capturado	Capturada	Sí	Buscar teléfono
Libre	Libre	No	Ir al oráculo
Libre	Fuera	No	Ir al oráculo
Capturado	Libre	Sí	Buscar teléfono
Libre	Capturada	Sí	Buscar teléfono

Codificación de la tabla:

- Personaje: libre = 0, capturado = 1, fuera = 2
- Smith: lejos = 0, cerca = 1
- Acción: Ir al oráculo = 0, Buscar teléfono = 1



Pregunta 2.1

- Dibuja la red neuronal lo más completamente posible, indicando donde se conectan las entradas y las salidas. Escribe también los pesos en el dibujo donde se correspondan, sabiendo que las tablas de pesos son las siguientes y teniendo en cuenta que, en esta red, por simplicidad, se considera que las neuronas no tienen sesgo.

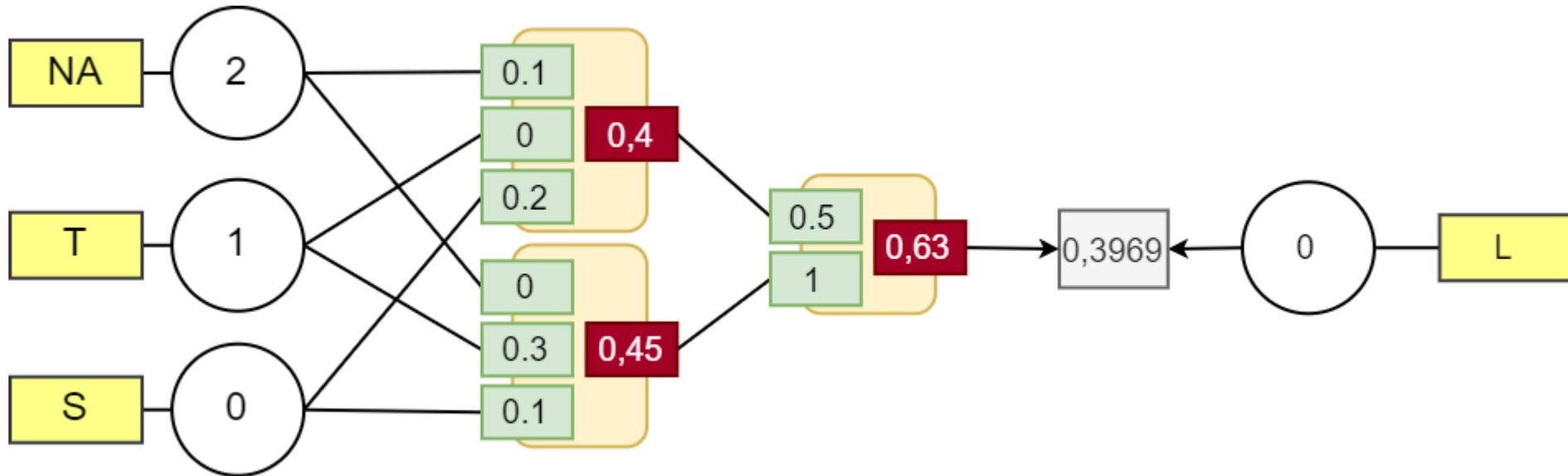
0.1	0
0	0.3
0.2	0.1

0.5
1

Pregunta 2.2

- ¿Cuál será la salida de la red neuronal construida en el apartado anterior, cuando recibe el primer ejemplo de la tabla? ¿Qué error se comete? Asumir que la función de activación sigmoide se calcula según la siguiente tabla:

x	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
sigmoide(x)	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8



DESARROLLO DE JUEGOS CON INTELIGENCIA ARTIFICIAL

Bloque II - Tema 2

Redes Neuronales – EJERCICIOS Soluciones



©2023 Autores
David María Arribas,
Sergio Hernández García,
Nicolás H. Rodríguez Uribe,
Alfredo Cuesta Infante

Algunos derechos reservados
Este documento se distribuye bajo la licencia
“Atribución-CompartirIgual 4.0 Internacional” de Creative Commons,
disponible en
<https://creativecommons.org/licenses/by-sa/4.0/deed.es>

David María
david.maria@urjc.es

Sergio Hernández
sergio.hernandez@urjc.es

Nicolás Rodríguez
nicolas.rodriguez@urjc.es



DESARROLLO DE JUEGOS CON INTELIGENCIA ARTIFICIAL

Bloque II - Tema 3

Árboles de decisión – Ejercicios Soluciones



Ejercicio 1: Selección de acciones

Un juego donde nuestro avatar es un francotirador tiene incorporada una IA que simula al centro estratégico, que nos sugiere acciones según nuestra actuación durante el juego. Las acciones que puede sugerir son: Lanzar granada, Dar rodeo, Avanzar recto. El centro estratégico ha monitorizado nuestra actuación en las ocasiones anteriores que hemos hecho alguna de estas tres acciones, almacenadas en la tabla de abajo.

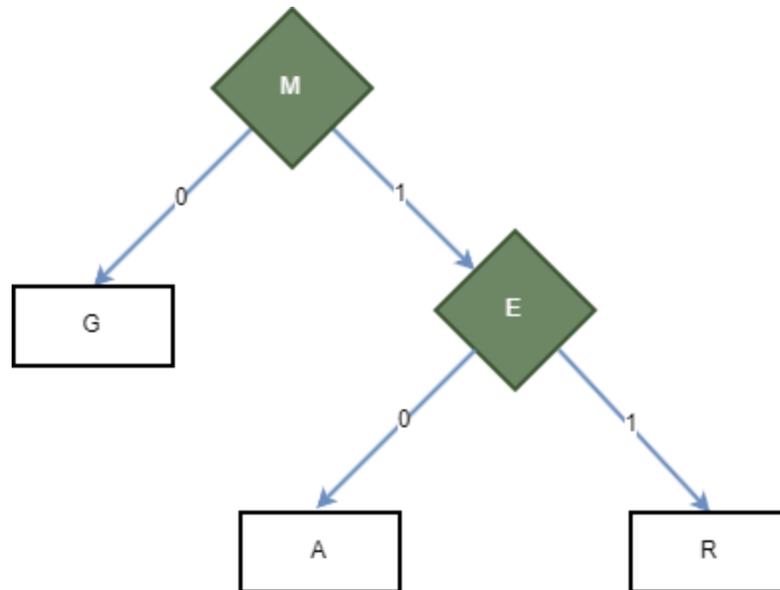


E	M	T	IA
15	10	Sí	G
18	50	Sí	R
5	39	No	A
6	12	No	R
2	100	Sí	A

- E** Número de **E**nemigos cerca
- M** **M**unición del Fusiñ
- T** **T**erreno Alto
- G** Lanzar **G**ranada
- R** Dar **R**odeo
- A** **A**vanzar Recto

Pregunta 1.1

- Aprender el árbol de decisión utilizando el algoritmo ID3, aplicando las siguientes normas para convertir los atributos en categorías:
 - $E = 1$ si el número de enemigos es mayor que 15, y 0 en caso contrario.
 - $M = 0$ si la munición es menor que el número de enemigos, y 1 en caso contrario.
 - $T = 1$ si estamos en terreno alto, y 0 en caso contrario

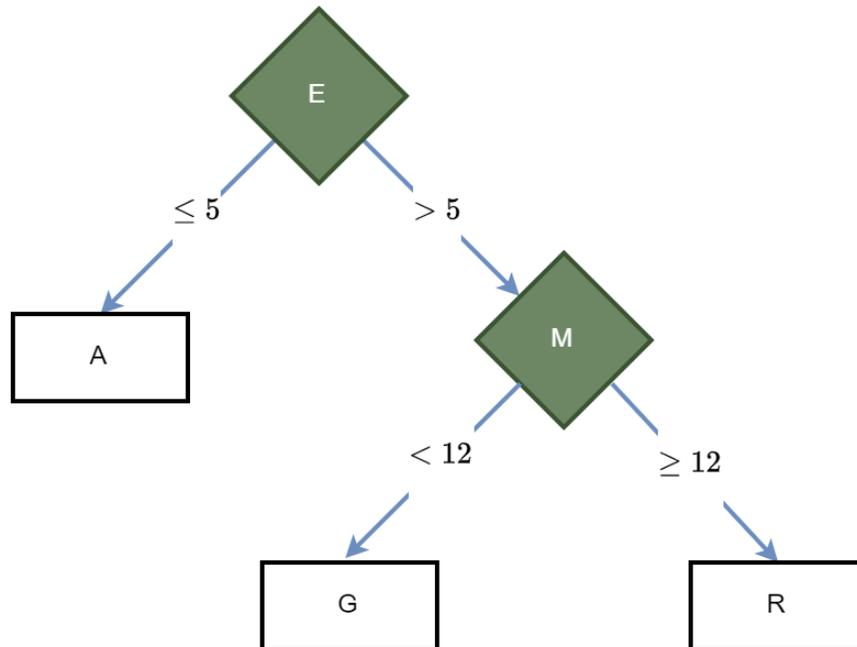


E	M	T	IA
15	10	Sí	G
18	50	Sí	R
5	39	No	A
6	12	No	R
2	100	Sí	A

E Número de **E**nemigos cerca
M **M**unición del Fusiñ
T **T**erreno Alto
G Lanzar **G**ranada
R Dar **R**odeo
A **A**vanzar Recto

Pregunta 1.2

- Aprender el árbol de decisión utilizando el algoritmo ID3, pero sin discretizar los atributos y aprendiendo una binarización en su lugar.



E	M	T	IA
15	10	Sí	G
18	50	Sí	R
5	39	No	A
6	12	No	R
2	100	Sí	A

E Número de **E**nemigos cerca
M **M**unición del Fusiñ
T Terreno Alto
G Lanzar **G**ranada
R Dar **R**odeo
A **A**vanzar Recto

Pregunta 1.3

- ¿Qué recomendaría el centro estratégico ante la situación de abajo según árbol del apartado anterior (1.2)?
 - 6 enemigos cerca, un cartucho de 20 municiones y terreno bajo
 - Se decide dar un rodeo (R)

E	M	T	IA
15	10	Sí	G
18	50	Sí	R
5	39	No	A
6	12	No	R
2	100	Sí	A

E Número de **E**nemigos cerca
M **M**unición del Fusiñ
T Terreno **A**lto
G Lanzar **G**ranada
R Dar **R**odeo
A **A**vanzar Recto

Pregunta 1.3

- Utilizar el algoritmo ID4 para incorporar al árbol resultante del ejercicio 1.1 la nueva fila dada a continuación.

E	M	T	IA
10	3	No	G

- No es necesario modificar el árbol. Como la munición es menor que el número de enemigos nos iremos por la rama de la izquierda, que lleva a una acción G (lanzar granada). Esta acción corresponde a la realizada para el nuevo ejemplo y por lo tanto no hay nada que debamos actualizar.

E	M	T	IA
15	10	Sí	G
18	50	Sí	R
5	39	No	A
6	12	No	R
2	100	Sí	A

E	Número de E nemigos cerca
M	M unición del Fusiñ
T	T erreno Alto
G	Lanzar G ranada
R	Dar R odeo
A	A vanzar Recto

Ejercicio 2: Selección de acciones

Nuestro equipo, Boston (verde), está jugando contra el equipo de la IA, Seattle (blanco). ¿Cuál será la decisión que tomará el jugador que lleva el balón (de blanco, a la izquierda de la canasta según se mira, dentro de la zona) a partir del árbol que se obtiene con ID3 y la tabla de datos dada abajo? Soluciona el problema categorizando el atributo T en tres posibles categorías y binarizando.

- Los datos para tomar la decisión son: D = 1, J = 3, T = 19

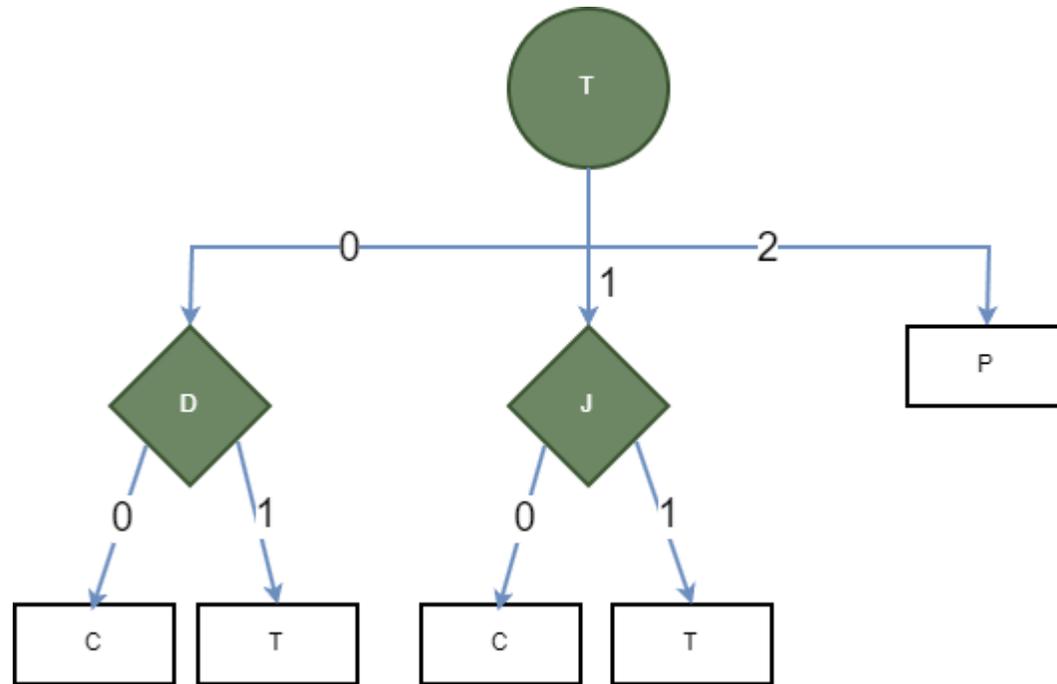


D	J	T	IA
1	1	8	T
0	1	28	P
0	1	8	C
0	5	15	C
1	3	29	P
0	0	17	T

- D Equipo contrario va por Delante
- J Jugadores contrarios en la zona
- T Tiempo de posesión
- P Pasar la pelota
- C Entrar a Canasta
- T Lanzar Triple

Ejercicio 2: Selección de acciones

- Solución obtenida categorizando T en los rangos Bajo = [0, 8], Medio = [9, 20], Alto = [21, 28] y binarizando J de forma que una clase es $J < 1$ y otra $J \geq 1$.



D	J	T	IA
1	1	8	T
0	1	28	P
0	1	8	C
0	5	15	C
1	3	29	P
0	0	17	T

D Equipo contrario va por Delante
J Jugadores contrarios en la zona
T Tiempo de posesión
P Pasar la pelota
C Entrar a Canasta
T Lanzar Triple

Ejercicio 3: Selección de acciones

En un juego de la saga StarWars, nuestro avatar es el piloto número 5 del escuadrón Negro de cazas Tie del Imperio. El resto del escuadrón son IA, incluido el líder (Negro 1). Al otro lado, la Alianza Rebelde ataca con el escuadrón Oro. Es indiferente quien gobierna esos cazas. Pueden ser jugadores en local, en red o NPCs. Negro 1 se comunica con nosotros a través de mensajes impresos en blanco en la parte superior de la pantalla para asignarnos los objetivos o blancos a los que atacar. Según cumplamos iremos ascendiendo en el escalafón del escuadrón e iremos estando al mando de otros cazas Tie, hasta llegar a ser líderes. Las decisiones del líder se aprenden a partir de las acciones pasadas del escuadrón Oro, cuyos efectos se resumen en la siguiente tabla.



D	C	B	Orden
12	1	0	L
1	2	1	S
10	5	3	S
10	3	1	L
9	2	0	L
2	1	3	S
8	5	1	L
2	4	3	L

- D Nivel de Daños nave principal
- C Número de Cazas enemigos
- B Bajas en nuestro escuadrón
- L Seguir al Líder
- S Seguir a un Superior asignado

Pregunta 3.1

- ¿Cuál será la orden que recibamos si la situación es que han caído dos cazas de nuestro escuadrón, el escuadrón Oro tiene 3 cazas y la nave nodriza tiene un nivel de daños igual a 7?

Nota: Siempre hay un líder y un superior. El líder es el piloto de mayor rango vivo ($1 > 2 > 3 > 4$). El superior es el piloto vivo de rango más próximo al nuestro.

D	C	B	Orden
12	1	0	L
1	2	1	S
10	5	3	S
10	3	1	L
9	2	0	L
2	1	3	S
8	5	1	L
2	4	3	L

- D** Nivel de **D**años nave principal
- C** Número de **C**azas enemigos
- B** **B**ajas en nuestro escuadrón
- L** Seguir al **L**íder
- S** Seguir a un **S**uperior asignado

Ejercicio 5: Selección de acciones

Nuestro estudio va a crear el próximo juego de Spider-Man. En una fase del juego Spiderman puede lograr un traje creado por Industrias Stark con una IA que le proporciona una defensa, endureciéndose, o un ataque, emitiendo energía en vez de tela de araña por la mano izquierda.

Construir un árbol de decisión para programar la IA que decide la ayuda a partir de la siguiente tabla de ejemplos, donde se mide la vitalidad de Spiderman, cuán favorable es el entorno para sus habilidades y el número de enemigos cerca. En caso de empate, la IA decide según el número de enemigos.

Vitalidad	Entorno	Enemigos	Ayuda
baja	favorable	3 o más	defensa
alta	desfavorable	3 o más	defensa
alta	favorable	menos de 3	defensa
baja	favorable	menos de 3	ataque
baja	desfavorable	3 o más	ataque

Ayuda.

$\log_2(x/y)$		y						
		1	2	3	4	5	6	7
x	1	0	-1	-1.6	-2	-2.3	-2.6	-2.8
	2	1	0	-0.6	-1	-1.3	-1.6	-1.8
	3	1.6	0.6	0	-0.4	-0.7	-1	-1.2
	4	2	1	0.4	0	-0.3	-0.6	-0.8
	5	2.3	1.3	0.7	0.3	0	-0.3	-0.5
	6	2.6	1.6	1	0.6	0.3	0	-0.2
	7	2.8	1.8	1.2	0.8	0.5	0.2	0

DESARROLLO DE JUEGOS CON INTELIGENCIA ARTIFICIAL

Bloque II - Tema 2

Árboles de decisión – Ejercicios Soluciones



©2023 Autores
David María Arribas,
Sergio Hernández García,
Nicolás H. Rodríguez Uribe,

Algunos derechos reservados
Este documento se distribuye bajo la licencia
"Atribución-CompartirIgual 4.0 Internacional" de Creative Commons, disponible en
<https://creativecommons.org/licenses/by-sa/4.0/deed.es>

David María
david.maria@urjc.es

Sergio Hernández
sergio.hernandez@urjc.es

Nicolás Rodríguez
nicolas.rodriguez@urjc.es



DESARROLLO DE JUEGOS CON INTELIGENCIA ARTIFICIAL

Bloque II - Tema 4

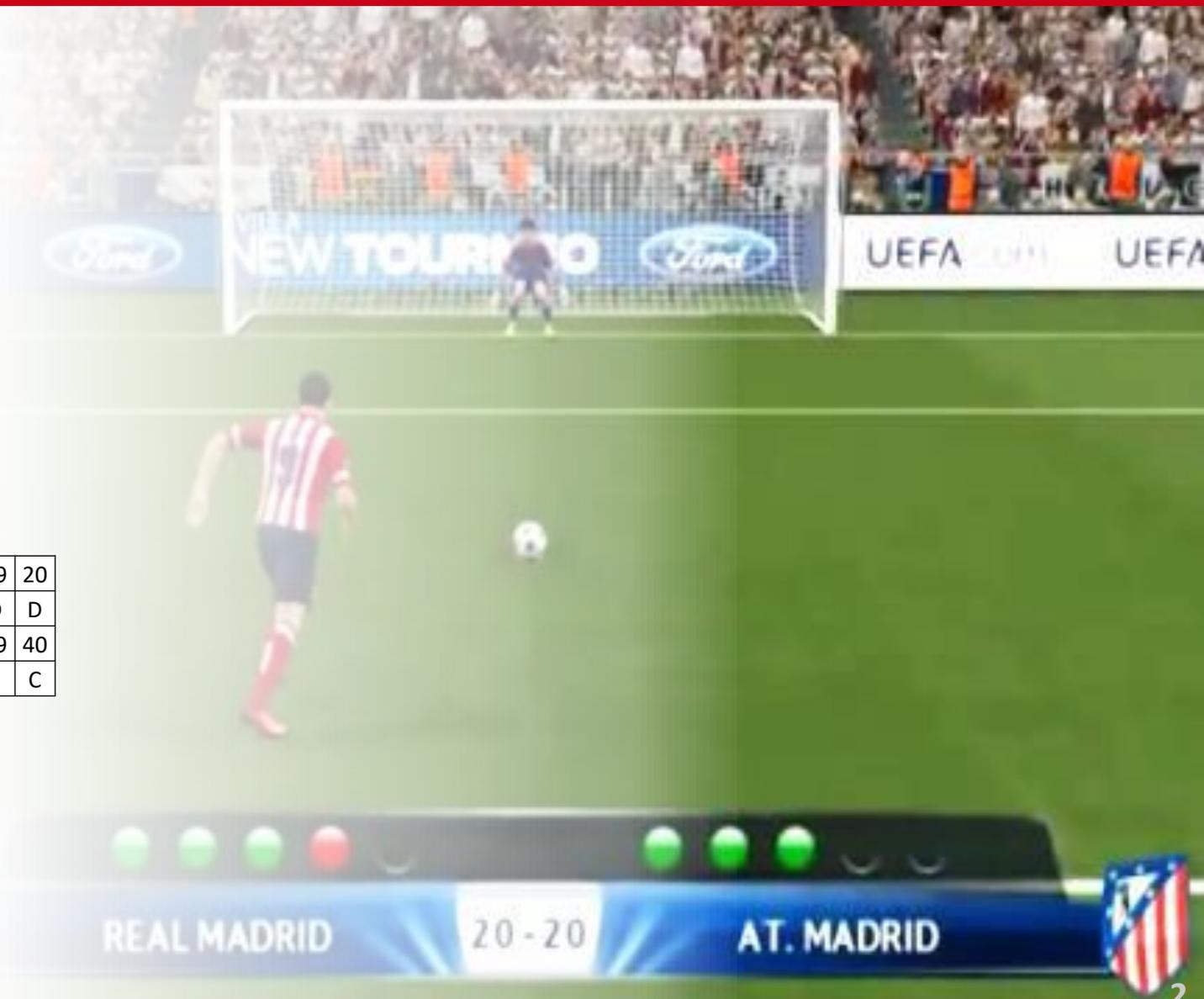
Predicción con N-gramas - EJERCICIOS



Ejercicio 1: predicción de acciones

En nuestro videojuego Futbol Pro 2024 una IA gobierna a los jugadores del Atlético de Madrid. Durante las partidas que ha jugado el jugador humano, se han grabado los últimos 40 penaltis recibidos en una tabla similar a la de abajo, donde I indica que el jugador (que maneja el portero) se lanzó a la izquierda, C que se quedó en el centro y D que se tiró a la derecha.

Lanzamiento	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Portero	I	C	I	C	C	I	D	I	D	D	I	C	D	I	I	D	C	I	D	D
Lanzamiento	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
Portero	D	C	D	C	I	D	I	I	C	I	C	C	C	I	I	I	C	I	I	C



Pregunta 1.1

- Genera las tablas de frecuencias correspondientes a los tres primeros N-gramas (N=1, N=2, N=3)

Unigrama

Patrón	Frecuencia
I	17
C	13
D	10

Bigrama

Patrón	Frecuencia
CC	3
CD	2
CI	7
DC	3
DD	3
DI	4
IC	7
ID	5
II	5

Trigrama

Patrón	Frecuencia	Patrón	Frecuencia
CCC	1	DIC	1
CCI	2	DID	1
CDC	1	DII	2
CDI	1	ICC	2
CIC	2	ICD	1
CID	3	ICI	3
CII	2	IDC	1
DCD	1	IDD	2
DCI	2	IDI	2
DDC	1	IIC	3
DDD	1	IID	1
DDI	1	III	1

Pregunta 1.2

- Si los 2 últimos lanzamientos han sido: **I C**
¿Hacia dónde NO deberá disparar el delantero en el siguiente penalti?
Resuelve el problema para los 3 ngramas escogiendo la probabilidad más alta.

Unigrama: *I*

$$P(I) = \frac{17}{40} = 0.43$$

$$P(C) = \frac{13}{40} = 0.32$$

$$P(D) = \frac{10}{40} = 0.25$$

Bigrama: *I*

$$P(I|C) = \frac{7}{12} = 0.58$$

$$P(C|C) = \frac{3}{12} = 0.25$$

$$P(D|C) = \frac{2}{12} = 0.17$$

Trigrama: *I*

$$P(I|IC) = \frac{3}{6} = 0.50$$

$$P(C|IC) = \frac{2}{6} = 0.33$$

$$P(D|IC) = \frac{1}{6} = 0.17$$

Pregunta 1.3

- Si los 2 últimos lanzamientos han sido: **I C**

¿Hacia dónde NO deberá disparar el delantero en el siguiente penalti?

Resuelve el problema realizando un muestreo aleatorio ponderado. El dado aleatorio devuelve 0.43

Unigrama: C

Bigrama: I

Trigrama: C

$$P(D) = \frac{10}{40} = 0.24 \rightarrow [0.00, 0.24)$$

$$P(D|C) = \frac{2}{12} = 0.17 \rightarrow [0.00, 0.17)$$

$$P(D|IC) = \frac{1}{6} = 0.17 \rightarrow [0.00, 0.17)$$

$$P(C) = \frac{13}{40} = 0.33 \rightarrow [0.24, 0.57)$$

$$P(C|C) = \frac{3}{12} = 0.25 \rightarrow [0.17, 0.42)$$

$$P(C|IC) = \frac{2}{6} = 0.33 \rightarrow [0.17, 0.50)$$

$$P(I) = \frac{17}{40} = 0.43 \rightarrow [0.57, 1.00)$$

$$P(I|C) = \frac{7}{12} = 0.58 \rightarrow [0.42, 1.00)$$

$$P(I|IC) = \frac{3}{6} = 0.50 \rightarrow [0.50, 1.00)$$

Ejercicio 2: predicción de acciones

En muchos juegos, para pasar de nivel hay que vencer a un enemigo mucho más potente que el resto. Para evitar que el jugador encuentre una secuencia de ataque que le venza fácilmente se implementa un predictor on-line que se entrena con los 3 n-gramas primeros. Así cuanto más tiempo dure la pelea más difícil será sorprenderle, pero este tendrá menos energía y así mantenemos nuestras posibilidades de vencer.

Nuestro avatar tiene un repertorio de 3 ataques normales más 1 ataque especial, que sólo puede lanzar si está 5 seg. sin ejecutar ningún ataque. Cada ataque requiere 1 seg., representando los **ataques normales como A, B y C, el ataque especial como S, y no atacar como Z.**

La secuencia almacenada se refleja abajo. El sistema usa los 3 primeros N-gramas (Unigrama, Bigrama, Trigrama) para aprender.

A C B C Z C Z Z Z Z Z S Z B C Z Z Z C A B A C A Z B B A B C Z Z Z Z Z B Z Z Z Z S A B B A Z Z Z



Pregunta 2.1

- ¿Cuántas filas máximas tendrán cada una de las tablas de frecuencias?

Unigrama: $S^N = 5^1$

Bigrama: $S^N = 5^2 = 25$

Trigrama: $S^N = 5^3 = 125$

Pregunta 2.2

- ¿Qué probabilidad hay de que la siguiente acción sea una A para cada patrón diferente?

Unigrama: 0.143

Bigrama:

Patrón	Frecuencia	Probabilidad
AA	0	0.000
BA	3	0.500
CA	2	0.333
SA	1	0.166
ZA	0	0.000

Trigrama:

Patrón	Frecuencia	Probabilidad
AAA	0	0.000
ABA	1	0.167
ACA	1	0.167
ASA	0	0.000
AZA	0	0.000
BAA	0	0.000
BBA	2	0.333
BCA	0	0.000
BSA	0	0.000
BZA	0	0.000
CAA	0	0.000
CBA	0	0.000
CCA	0	0.000
CSA	0	0.000
CZA	0	0.000
SAA	0	0.000
SBA	0	0.000
SCA	0	0.000
SSA	0	0.000
SZA	0	0.000
ZAA	0	0.000
ZBA	0	0.000
ZCA	1	0.167
ZSA	1	0.167
ZZA	0	0.000

Pregunta 2.2

- ¿Cuál es nuestra acción más probable en el siguiente segundo?

Últimos sucesos: z z

Unigrama

Patrón	Frecuencia	Probabilidad
A	7	0.143
B	9	0.184
C	7	0.143
S	2	0.041
Z	24	0.490

Bigrama

Patrón	Frecuencia	Probabilidad
ZB	3	0.130
ZC	2	0.087
ZS	2	0.087
ZZ	16	0.696

Trigrama

Patrón	Frecuencia	Probabilidad
ZZB	1	0.045
ZZC	1	0.045
ZZS	2	0.091
ZZZ	11	0.500

Pregunta 2.3

- Discutir si las probabilidades obtenidas son plausibles según las condiciones del ataque S .

No se podría producir el ataque S ya que los tamaños de ventana elegidos no permiten registrar el patrón necesario para que este suceso se produzca (5 Z seguidos). Sería necesario un 6-grama para poder registrar los eventos que permitan ejecutar el ataque S .

Ejercicio 2: predicción de acciones

En un combate aéreo la IA registra nuestros movimientos cuando hemos capturado un avión enemigo como blanco en nuestro sistema de seguimiento.

Cada cinco segundos la IA convierte nuestros movimientos en alguna de las siguientes tres “maniobras” de persecución: {A, B, C}.

La secuencia de maniobras se encuentra a continuación.

A B A C B A B A C A B A B B A A B A B C C A C A B C A C A B B B A C B A B A C



Pregunta 3.1

- Crear la **tabla de probabilidades** para un predictor 3-grama según se va produciendo la secuencia de maniobras.

Patrón	Recuento	Probabilidad
AAB	1	1.000
ABA	5	0.556
ABB	2	0.222
ABC	2	0.222
ACA	3	0.600
ACB	2	0.400
BAA	1	0.111
BAB	4	0.444
BAC	4	0.444
BBA	2	0.667
BBB	1	0.333
BCA	1	0.500
BCC	1	0.500
CAB	3	0.600
CAC	2	0.400
CBA	2	1.000
CCA	1	1.000

Pregunta 3.2

- ¿Cuál es el 3-grama que más aparece?

ABA

- ¿Cuál es su probabilidad?

$$P(ABA) = \frac{5}{37} = 0.135$$

Pregunta 3.3

- ¿Qué maniobra predecirá la IA a continuación?

Últimos movimientos: A C

$$P(A|AC) = \frac{3}{5} = 0.600$$

$$P(B|AC) = \frac{2}{5} = 0.400$$

La predicción es: A

Ejercicio 4: predicción de acciones

Nuestro estudio va a crear el próximo juego de Spider-Man. En un nivel el villano será el Dr. Octopus, que lleva 4 brazos mecánicos a su espalda lo que le da fuerza, agilidad, rapidez y movilidad. Pero no deja de ser un humano que acusa el cansancio y los golpes, y por tanto pierde vida a un ritmo 'normal'. Spider-Man tiene el mismo problema, por eso la mejor manera de luchar es moverse mucho y golpear de vez en cuando.

Para diseñar la IA del Dr. Octopus durante las peleas con Spider-Man, se implementa un 2-grama, un 3-grama y un 4-grama.

Las posibles acciones son (S = saltar, G = golpear, I = izquierda, D = derecha). El histórico de acciones realizadas por Spider-Man se encuentra a continuación

S D I D I S I S I D D G S D I D I D G S D I D I D I D S I G S G S D I S I G D S D I



Pregunta 4.1

- ¿Cuál es la estimación para el próximo movimiento? ¿Con qué probabilidad?

I		
Patrón	Frecuencia	Probabilidad
ID	7	0.583
IG	2	0.167
IS	3	0.250

D I		
Patrón	Frecuencia	Probabilidad
DID	6	0.750
DIS	2	0.250

S D I		
Patrón	Frecuencia	Probabilidad
SDID	3	0.750
SDIS	1	0.250

DESARROLLO DE JUEGOS CON INTELIGENCIA ARTIFICIAL

Bloque II - Tema 4

Predicción con N-gramas - EJERCICIOS



©2023 Autores
David María Arribas,
Sergio Hernández García,
Nicolás H. Rodríguez Uribe,

Algunos derechos reservados
Este documento se distribuye bajo la licencia
"Atribución-CompartirIgual 4.0 Internacional" de Creative Commons, disponible en
<https://creativecommons.org/licenses/by-sa/4.0/deed.es>

David María
david.maria@urjc.es

Sergio Hernández
sergio.hernandez@urjc.es

Nicolás Rodríguez
nicolas.rodriguez@urjc.es



DESARROLLO DE JUEGOS CON INTELIGENCIA ARTIFICIAL

Bloque II - Tema 5

Q-Learning-EJERCICIOS Soluciones



Ejercicio 1

En un juego de mafiosos nuestro avatar (un sicario) está huyendo a través de la ciudad en coche. La policía (una IA) sale a su encuentro en diferentes puntos para detenernos. Inicialmente intenta tres acciones básicas: bloquearnos el paso, perseguirnos o chocar. Queremos que, según avance el juego, la policía sea más eficiente gracias a la experiencia ganada contra nuestra habilidad al volante. El comportamiento de la policía se aprende mediante Q-learning. Para ello los estados del juego se modelan de forma que contengan la siguiente información:

- Velocidad del coche perseguido: {Espacio, Medio, Alto}
- Carriles en la calzada: {1, 2, 3 o más}
- Tramo de la calzada: {Recto, Curva}
- Daños del coche patrulla: {Leves, Graves}



Pregunta 1.1

- ¿Cuántos estados diferentes hay?
- $3 \cdot 3 \cdot 2 \cdot 2 = 36$

Pregunta 1.2

- Suponiendo las siguientes tablas de valores Q, recompensa (r) y transiciones a estado siguiente S'. Calcular el nuevo valor Q si estamos en el estado [X] y seleccionamos la acción [K], donde
 - $X = 1 +$ el resto de dividir tu DNI entre el número de estados diferentes,
 - $K = 1 +$ el resto de dividir tu DNI entre el número de acciones diferentes.
 - Los controles toman los siguientes valores $\alpha = 0,3$ y $\gamma = 0,8$.

Tabla Q:

		Estado actual S																																			
Q		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
Acción	B	6	53	34	49	15	14	5	1	12	4	1	36	49	41	33	50	4	57	16	25	51	5	23	52	38	34	33	5	21	3	54	58	3	57	29	29
	P	31	31	31	48	44	59	47	6	29	2	37	2	51	7	53	34	59	39	57	7	12	38	44	41	47	23	45	58	34	20	1	10	2	6	54	15
	C	10	34	31	23	23	57	38	25	56	16	9	14	56	13	39	30	12	58	4	33	10	22	48	10	42	9	49	25	59	59	49	24	7	22	16	30

Tabla de recompensas:

		Estado actual S																																				
r		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	
Acción	B	100	0	0	0	100	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	C	100	0	0	0	100	0	0	0	100	0	0	0	100	0	0	0	100	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabla de transiciones:

		Estado actual S																																			
S'		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
Acción	B	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
	P	13	14	3	4	17	18	19	20	21	22	23	24	25	26	15	16	29	30	31	32	33	34	35	36	25	26	27	28	29	30	31	32	33	34	35	36
	C	1	2	3	3	5	6	3	3	9	10	3	3	1	14	3	3	5	18	3	3	9	22	3	3	13	26	3	3	17	30	3	3	21	34	3	3

Pregunta 1.2

Tabla Q:

		Estado actual S																																			
Q		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
Acción	B	6	53	34	49	15	14	5	1	12	4	1	36	49	41	33	50	4	57	16	25	51	5	23	52	38	34	33	5	21	3	54	58	3	57	29	29
	P	31	31	31	48	44	59	47	6	29	2	37	2	51	7	53	34	59	39	57	7	12	38	44	41	47	23	45	58	34	20	1	10	2	6	54	15
	C	10	34	31	23	23	57	38	25	56	16	9	14	56	13	39	30	12	58	4	33	10	22	48	10	42	9	49	25	59	59	49	24	7	22	16	30

Tabla de recompensas:

		Estado actual S																																				
r		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	
Acción	B	100	0	0	0	100	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	C	100	0	0	0	100	0	0	0	100	0	0	0	100	0	0	0	100	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabla de transiciones:

		Estado actual S																																			
S'		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
Acción	B	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
	P	13	14	3	4	17	18	19	20	21	22	23	24	25	26	15	16	29	30	31	32	33	34	35	36	25	26	27	28	29	30	31	32	33	34	35	36
	C	1	2	3	3	5	6	3	3	9	10	3	3	1	14	3	3	5	18	3	3	9	22	3	3	13	26	3	3	17	30	3	3	21	34	3	3

Tomando $X = 19$ y $K = B$:

$$Q'(19, B) = (1 - \alpha)Q(19, B) + \alpha(r + \gamma Q(7, P)) = 0,7 \cdot 16 + 0,3(0 + 0,8 \cdot 47) = 11,2 + 11,28 = 22,48$$

Pregunta 1.3

- Continuar actualizando la tabla Q hasta que lleguemos al estado 1, 5 o 9; o bien al realizar 5 iteraciones del algoritmo. Suponer que siempre elegimos la acción que tiene mayor valor Q.

- Continuando el apartado anterior $X = 7$ y $K = P$:

$$Q'(7, P) = (1-\alpha)Q(7, P) + \alpha(r + \gamma Q(19, P)) = 0,7 \cdot 47 + 0,3(0 + 0,8 \cdot 57) = 32,9 + 13,68 = 46,58$$

- $X = 19$ y $K = P$:

$$Q'(19, P) = (1-\alpha)Q(19, P) + \alpha(r + \gamma Q(31, B)) = 0,7 \cdot 57 + 0,3(0 + 0,8 \cdot 54) = 39,9 + 12,96 = 52,86$$

- $X = 31$ y $K = B$:

$$Q'(31, B) = (1-\alpha)Q(31, B) + \alpha(r + \gamma Q(19, P)) = 0,7 \cdot 54 + 0,3(0 + 0,8 \cdot 52,86) = 37,8 + 12,68 = 50,49$$

- $X = 19$ y $K = P$:

$$Q'(19, P) = (1-\alpha)Q(19, P) + \alpha(r + \gamma Q(31, B)) = 0,7 \cdot 52,86 + 0,3(0 + 0,8 \cdot 50,49) = 37,0 + 12,12 = 49,12$$

- $X = 31$ y $K = B$:

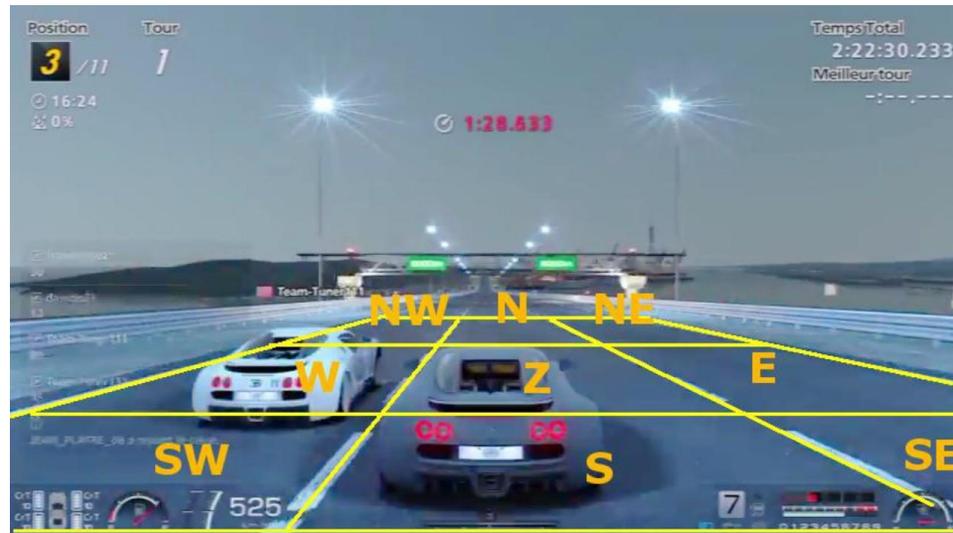
$$Q'(31, B) = (1-\alpha)Q(31, B) + \alpha(r + \gamma Q(19, P)) = 0,7 \cdot 50,49 + 0,3(0 + 0,8 \cdot 49,12) = 35,34 + 11,78 = 47,13$$

Ejercicio 2

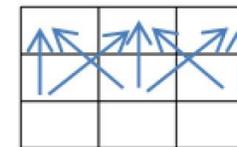
En un juego de carreras queremos crear la inteligencia del coche NPC mediante Q-learning. Para ello modelamos el área alrededor del coche como una malla 3×3 donde el coche siempre está en la fila central, es decir puede estar en las casillas W, Z o E. En cambio nuestro coche puede estar en cualquiera, salvo en la que ocupa el coche NPC, o incluso no estar (puede ir muy por delante o muy por detrás). El coche NPC se puede mover hacia delante o en diagonal, pero no puede pasar del carril izquierdo al derecho ni viceversa sin pasar por el central. Cuando el coche avanza la malla retrocede para que la celda que ocupará nuestro coche sea W, Z o E. El avance nuestro coche se produce después, pero como la pantalla se refresca tan rápidamente no se aprecia ningún retardo.

Por ejemplo, en la imagen de arriba el coche NPC (blanco) está en W. Por tanto puede elegir entre NW o N.

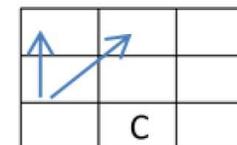
- Si elige NW entonces la celda que ocupará en el instante siguiente será W de nuevo.
- Si elige N entonces la celda que ocupará en el instante siguiente será Z.
- Si elige NE entonces el episodio de entrenamiento termina ya que no puede desplazarse a esa celda.



Movimientos posibles

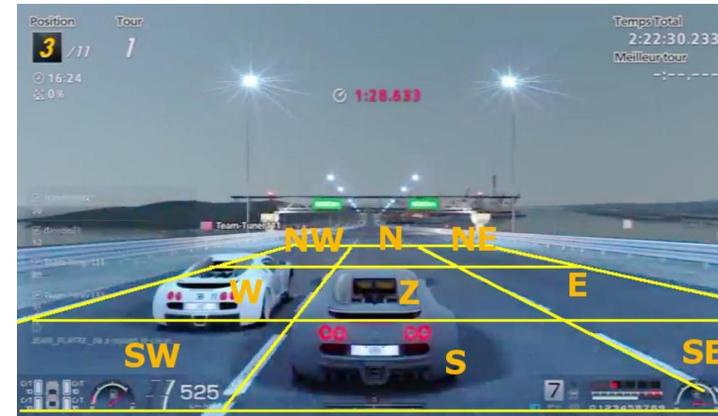


En la imagen

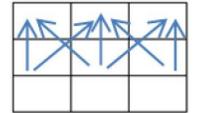


Pregunta 2.1

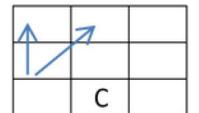
- Suponiendo que el estado se forma con la posición del coche NPC y nuestro coche ¿cuántos estados diferentes hay?
- 3 posibles posiciones del NPC, 9 del jugador y 3 estados donde ambos coches ocupan la misma casilla y que no son posibles.
- $3 \cdot 9 - 3 = 24$



Movimientos posibles



En la imagen



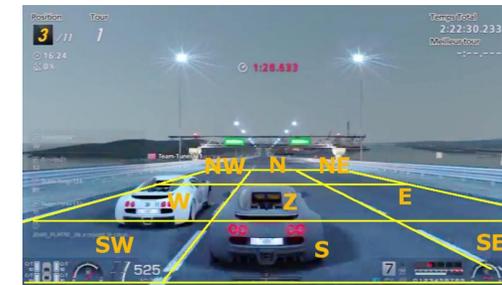
Pregunta 2.2

- Proponer una tabla (o función) de recompensa suponiendo que el objetivo del coche NPC es ir siempre (o el máximo tiempo posible) delante de nuestro coche bloqueándolo y que nosotros mantenemos la dirección.

r		Acción			
NPC	YO	S	NW	N	NE
W	NW	1	-1	0	-1
W	N	2	0	-1	-1
W	NE	3	0	0	-1
W	Z	4	0	100	-1
W	E	5	0	0	-1
W	SW	6	100	0	-1
W	S	7	0	100	-1
W	SE	8	0	0	-1
Z	NW	9	-1	0	0
Z	N	10	0	-1	0
Z	NE	11	0	0	-1
Z	W	12	100	0	0
Z	E	13	0	0	100
Z	SW	14	100	0	0
Z	S	15	0	100	0
Z	SE	16	0	0	100
E	NW	17	-1	0	0
E	N	18	-1	-1	0
E	NE	19	-1	0	-1
E	W	20	-1	0	0
E	Z	21	-1	100	0
E	SW	22	-1	0	0
E	S	23	-1	100	0
E	4	24	-1	0	100

nw	n	ne
w	z	e
sw	s	se

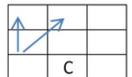
1 yo npc	2 npc yo	3 npc yo	4 npc yo	5 npc yo	6 npc yo	7 npc yo	8 npc yo
9 yo npc	10 npc yo	11 npc yo	12 yo npc	13 npc yo	14 yo npc	15 npc yo	16 npc yo
17 yo npc	18 npc yo	19 npc yo	20 yo npc	21 yo npc	22 yo npc	23 yo npc	24 npc yo



Movimientos posibles

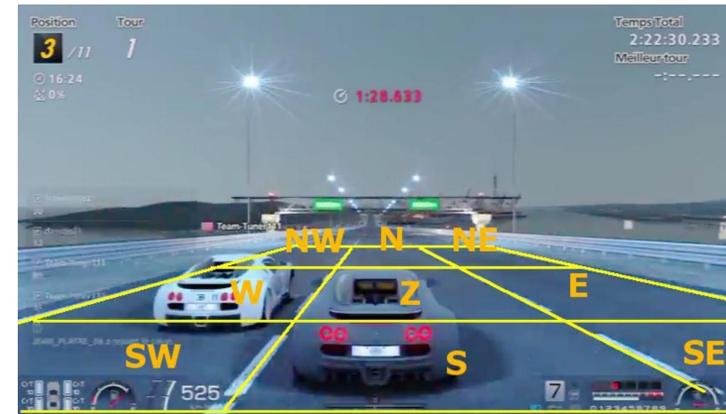


En la imagen

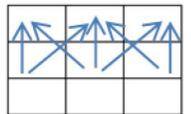


Pregunta 2.3

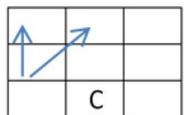
- ¿Cuál será el nuevo valor Q en la escena de la imagen para cada movimiento posible del coche NPC? Todos los valores de la tabla Q se acaban de inicializar a valor cero.
- Datos: $\alpha = 0,3$; $\gamma = 0,8$
- $Q'(\{W, S\}, NW) = (1-\alpha)Q(\{W, S\}, NW) + \alpha(r + \gamma \max_{a'} Q(s', a')) = 0,7 \cdot 0 + 0,3(0 + 0,8 \cdot 0) = 0$
- $Q'(\{W, S\}, N) = (1-\alpha)Q(\{W, S\}, N) + \alpha(r + \gamma \max_{a'} Q(s', a')) = 0,7 \cdot 0 + 0,3(100 + 0,8 \cdot 0) = 30$
- $Q'(\{W, S\}, NE) = (1-\alpha)Q(\{W, S\}, NE) + \alpha(r + \gamma \max_{a'} Q(s', a')) = 0,7 \cdot 0 + 0,3((-1) + 0,8 \cdot 0) = -0,3$



Movimientos posibles



En la imagen



Ejercicio 3

Nuestro estudio está desarrollando un juego ambientado en el oeste americano donde nuestro jugador es un sheriff y su ayudante es un rastreador nativo. Durante una fase del juego el rastreador nos lleva hasta el escondite de los bandidos, que está en la casilla (5.5) del mapa de la Figura de abajo a la derecha. El rastreador es una IA que ha sido entrenada con Q-learning, cuyo resultado es la tabla Q de la Figura de abajo a la izquierda.

Q	1.1	1.2	1.3	1.4	1.5	2.1	2.2	2.3	2.4	2.5	3.1	3.2	3.3	3.4	3.5	4.1	4.2	4.3	4.4	4.5	5.1	5.2	5.3	5.4	5.5
N	0	0	0	0	0	12	10	15	22	8	7	8	3	1	7	2	8	7	4	36	54	47	7	19	0
E	11	6	12	0	0	4	19	23	18	0	16	15	24	11	0	7	15	11	20	0	48	23	21	0	0
S	8	13	6	6	6	21	20	19	26	5	7	4	11	5	8	5	12	35	14	64	0	0	0	0	0
O	0	12	9	11	7	0	12	22	22	6	0	9	5	9	2	0	8	29	16	30	0	18	11	3	0

1.1	1.2	1.3	1.4	1.5
2.1	2.2	2.3	2.4	2.5
3.1	3.2	3.3	3.4	3.5
4.1	4.2	4.3	4.4	4.5
5.1	5.2	5.3	5.4	5.5

Pregunta 3.1

- Suponiendo que empezamos la búsqueda en la esquina inferior izquierda, casilla (5.1), ¿por qué camino nos llevará el rastreador?

Q	1.1	1.2	1.3	1.4	1.5	2.1	2.2	2.3	2.4	2.5	3.1	3.2	3.3	3.4	3.5	4.1	4.2	4.3	4.4	4.5	5.1	5.2	5.3	5.4	5.5
N	0	0	0	0	0	12	10	15	22	8	7	8	3	1	7	2	8	7	4	36	54	47	7	19	0
E	11	6	12	0	0	4	19	23	18	0	16	15	24	11	0	7	15	11	20	0	48	23	21	0	0
S	8	13	6	6	6	21	20	19	26	5	7	4	11	5	8	5	12	35	14	64	0	0	0	0	0
O	0	12	9	11	7	0	12	22	22	6	0	9	5	9	2	0	8	29	16	30	0	18	11	3	0

1.1	1.2	1.3	1.4	1.5
2.1	2.2	2.3	2.4	2.5
3.1	3.2	3.3	3.4	3.5
4.1	4.2	4.3	4.4	4.5
5.1	5.2	5.3	5.4	5.5

Diagram illustrating a search path on a 5x5 grid. The path starts at (5.1) and moves right to (5.2), then up to (4.1), right to (4.2), right to (4.3), down to (5.3), up to (4.4), and right to (4.5). The starting cell (5.1) is green, and the target cell (5.5) is red.

Pregunta 3.2

- Suponiendo que la recompensa por tomar una acción que lleve al escondite es 100, y que el rastreador decide ir al Este (E) desde la casilla (5.4) ¿qué celda de la tabla Q cambia?, ¿cuál es el nuevo valor?
- Los parámetros son $\alpha = 0,3$ y $\gamma = 0,8$

Q	1.1	1.2	1.3	1.4	1.5	2.1	2.2	2.3	2.4	2.5	3.1	3.2	3.3	3.4	3.5	4.1	4.2	4.3	4.4	4.5	5.1	5.2	5.3	5.4	5.5
N	0	0	0	0	0	12	10	15	22	8	7	8	3	1	7	2	8	7	4	36	54	47	7	19	0
E	11	6	12	0	0	4	19	23	18	0	16	15	24	11	0	7	15	11	20	0	48	23	21	0	0
S	8	13	6	6	6	21	20	19	26	5	7	4	11	5	8	5	12	35	14	64	0	0	0	0	0
O	0	12	9	11	7	0	12	22	22	6	0	9	5	9	2	0	8	29	16	30	0	18	11	3	0

1.1	1.2	1.3	1.4	1.5
2.1	2.2	2.3	2.4	2.5
3.1	3.2	3.3	3.4	3.5
4.1	4.2	4.3	4.4	4.5
5.1	5.2	5.3	5.4	5.5

Se modifica la celda referente al estado 5.4 y la acción E.

$$Q'(\{5,4\},E) = (1-\alpha)Q(\{5,4\},E) + \alpha(r + \gamma \max_{a'} Q(\{5,5\}, a')) = 0,7 \cdot 0 + 0,3(100 + 0,8 \cdot 0) = 30$$

DESARROLLO DE JUEGOS CON INTELIGENCIA ARTIFICIAL

Bloque II - Tema 2

Q-Learning-EJERCICIOS

Soluciones



©2023 Autores
David María Arribas,
Sergio Hernández García,
Nicolás H. Rodríguez Uribe,

Algunos derechos reservados
Este documento se distribuye bajo la licencia
"Atribución-CompartirIgual 4.0 Internacional" de Creative Commons, disponible en
<https://creativecommons.org/licenses/by-sa/4.0/deed.es>

David María
david.maria@urjc.es

Sergio Hernández
sergio.hernandez@urjc.es

Nicolás Rodríguez
nicolas.rodriguez@urjc.es

