

PED 1

Inteligencia Artificial

Javier Martínez Coll

20236575W

C.A: Valencia-Gandía

Ejercicio 1

Primeramente debemos tener en cuenta la cantidad de operadores que podemos tener para cambiar de un estado a otro, en este caso cada disco tiene la opción de moverse entre 3 torres. Por lo tanto los operadores quedarían de la siguiente manera:

Mover Disc1 a T1	Mover Disc1 a T2	Mover Disc1 a T3
Mover Disc2 a T1	Mover Disc2 a T2	Mover Disc2 a T3
Mover Disc3 a T1	Mover Disc3 a T2	Mover Disc3 a T3

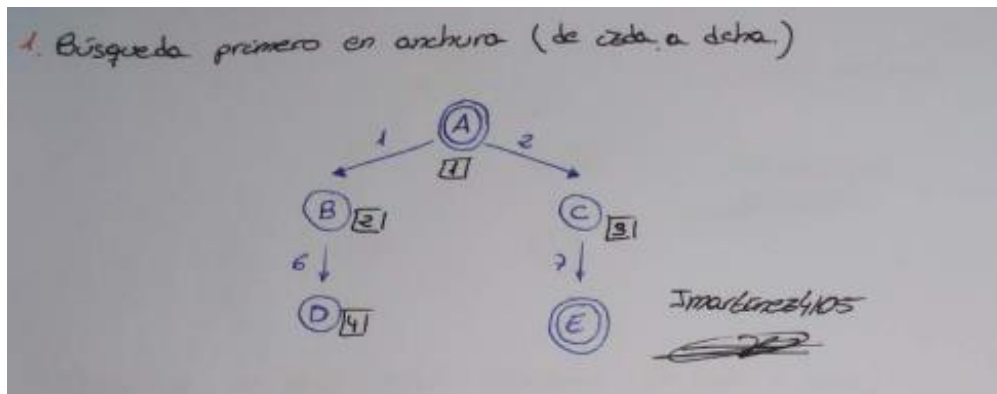
Por tanto en número de operadores máximos que podemos tener es 9.

Por otro lado, el número de estados posibles asciende a 27 estados ($3 \times 3 \times 3$). Debemos de tener en cuenta que hemos contabilizado también los estados erróneos o que se encuentren dentro de las restricciones del problema.

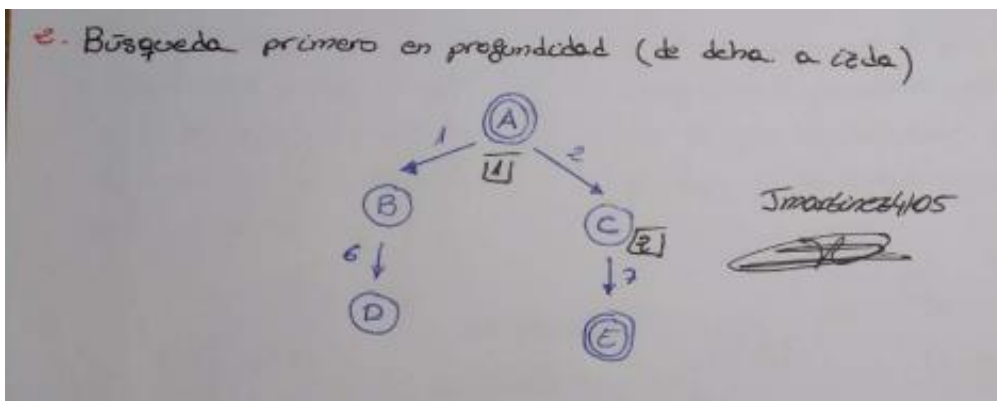
En cuanto a la nomenclatura de los elementos usados en la tabla, debemos tener en cuenta que el Disco 1 es el más pequeño, mientras que el Disco 3 es el más grande y T1 es la torre inicial, mientras que T3 es la torre del estado final.

Ejercicio 2

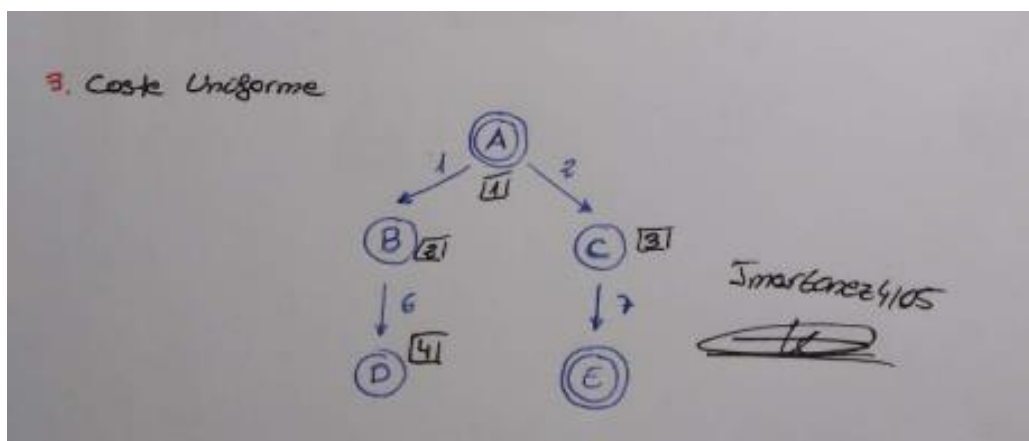
1. Búsqueda Primero en Anchura (de izquierda a derecha)



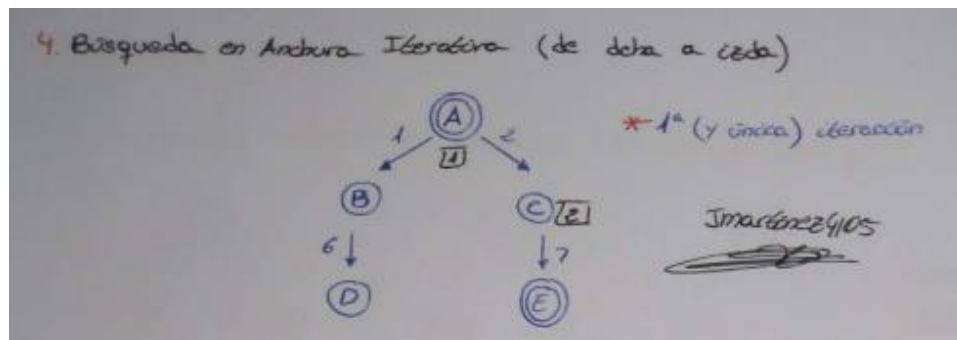
2. Búsqueda Primero en Profundidad (de derecha a izquierda)



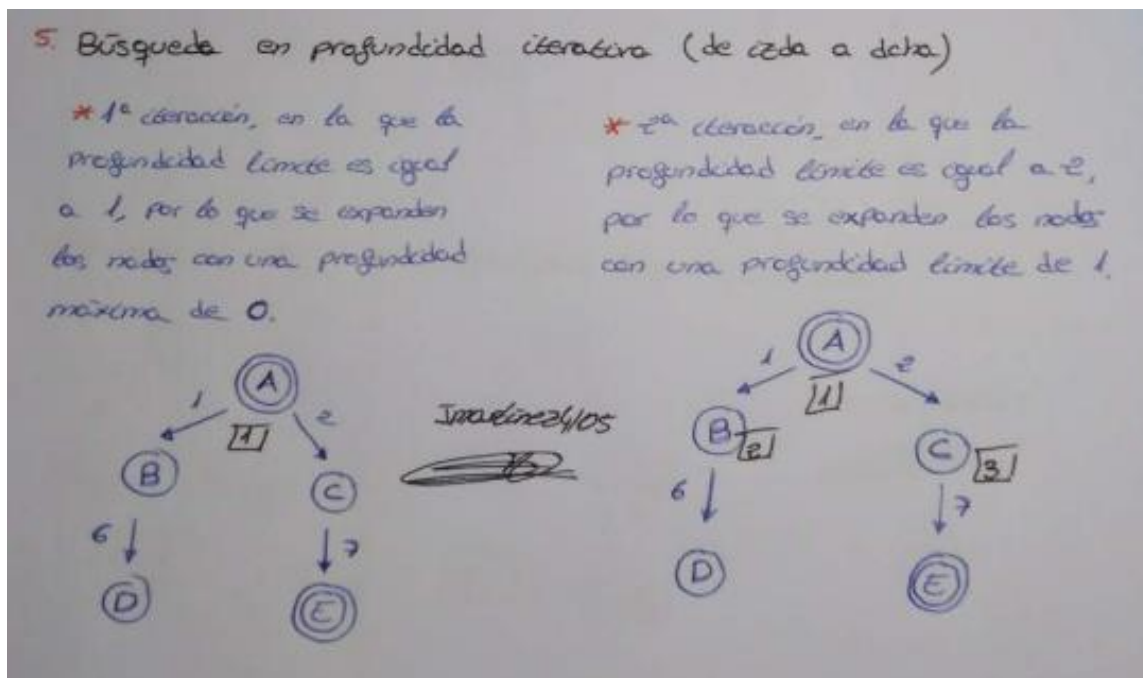
3. Búsqueda de Coste Uniforme



4. Búsqueda en Anchura Iterativa (de derecha a izquierda)



5. Búsqueda en Profundidad Iterativa (de izquierda a derecha)



Ejercicio

1. Búsqueda Primero en Anchura (de izquierda a derecha)

Antes de sacar A: {A}

Antes de sacar B: {B, C}

Antes de sacar C: {C, D}

Antes de sacar D: {D, E}

Antes de sacar E: {E}

Meta encontrada

Podemos observar que la tabla de ABIERTA en recorrido en anchura se comporta como una cola.

2. Búsqueda Primero en Profundidad (de derecha a izquierda)

Antes de sacar A: {A}

Antes de sacar C: {C, B}

Antes de sacar E: {E, B}

Meta encontrada

Podemos observar que la tabla ABIERTA en recorrido en profundidad se comporta como una cola.

3. Búsqueda de Coste Uniforme

Antes de sacar A: {A}

Antes de sacar B: {B (1), C (2)}

Antes de sacar C: {C (2), D (7)}

Antes de sacar D: {D (7), E (9)}

Antes de sacar E: {E (9)}

Meta encontrada

4. Búsqueda en Anchura Iterativa (de derecha a izquierda)

Iteración 1 (cada expansión de un nodo padre genera como máximo un hijo):

Antes de sacar A: {A}

Antes de sacar C: {C}

Antes de sacar E: {E}

Meta encontrada

5. Búsqueda en Profundidad Iterativa (de izquierda a derecha)

Iteración 1 (La profundidad límite es igual a 1):

Antes de sacar A: {A}

Antes de sacar B: {B, C}

Antes de sacar C: {C}

No expandimos los nodos B y C por coincidir con la profundidad límite.

Antes de sacar A: {A}

Antes de sacar B: {B, C}

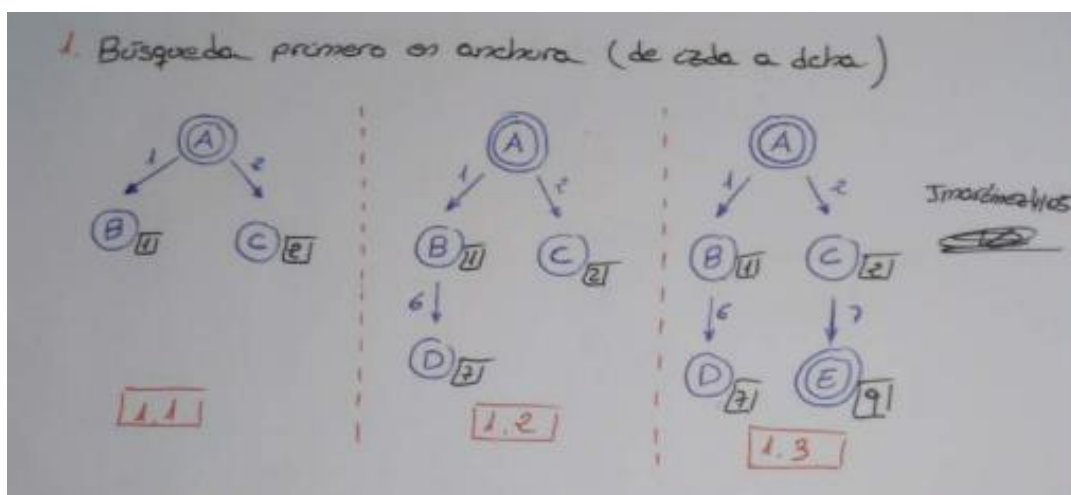
Antes de sacar D: {D, C}

Antes de sacar C: {C}

Antes de sacar E: {E} **Meta encontrada**

Ejercicio 4

1. Búsqueda Primero en Anchura (de izquierda a derecha)

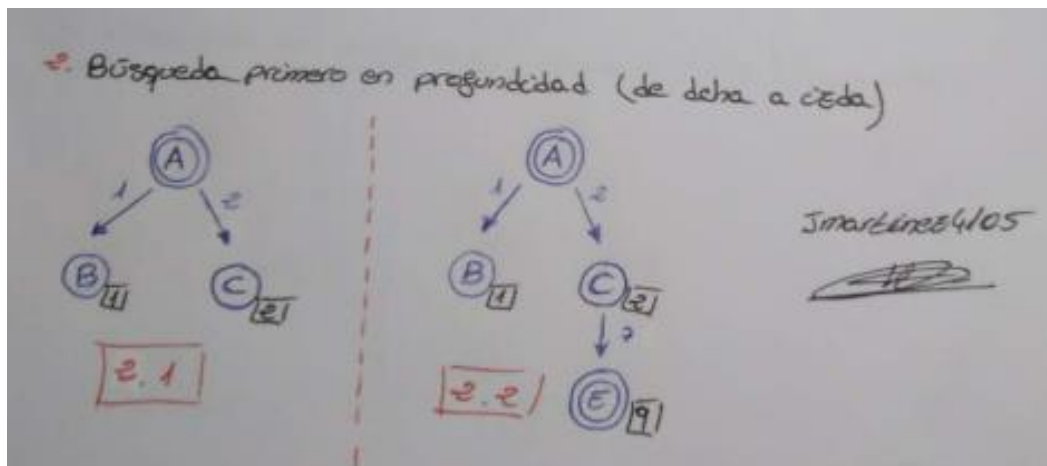


En primer lugar expandimos el nodo inicial, generando sus nodos hijos. Desde cada nodo hijo trazamos un nodo ascendente a su nodo padre. Al lado de cada nodo hijo indicamos el coste desde dicho nodo al nodo inicial.

En 1.2 expandimos el nodo B.

En 1.3 expandimos el nodo C, a continuación intentamos expandir el nodo D, pero este no contiene hijos, así que expandimos el nodo E y llegamos a que es un nodo meta. El coste del camino hallado hasta la meta es de 9.

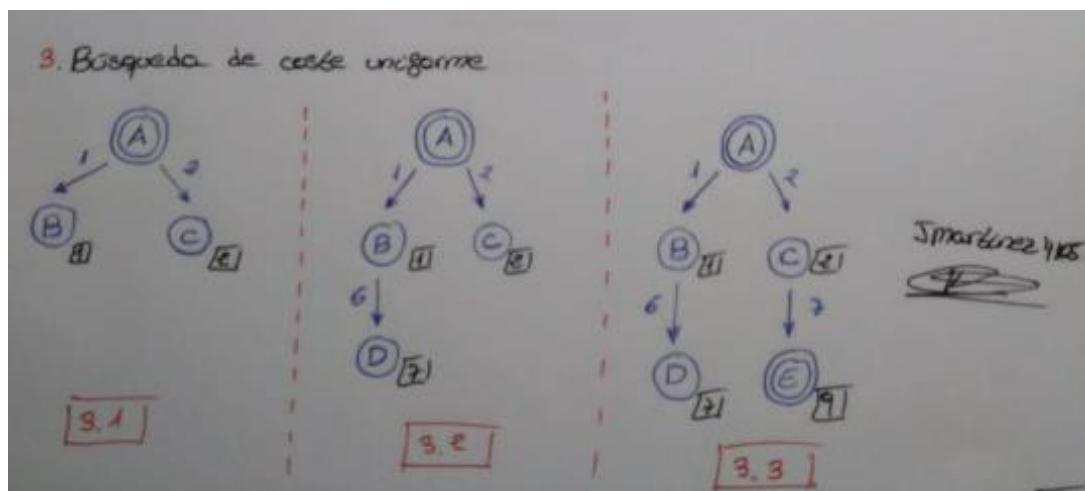
2. Búsqueda Primero en Profundidad (de derecha a izquierda)



En primer lugar expandimos el nodo inicial, generando sus nodos hijos. Desde cada nodo hijo trazamos un nodo ascendente a su nodo padre. Al lado de cada nodo hijo indicamos el coste desde dicho nodo al nodo inicial, como ocurre en 2.1.

En 2.2 expandimos en nodo C y a su vez el nodo E, al ser este último el nodo meta hemos hallado el final. El coste del camino es de 9.

3. Búsqueda de Coste Uniforme

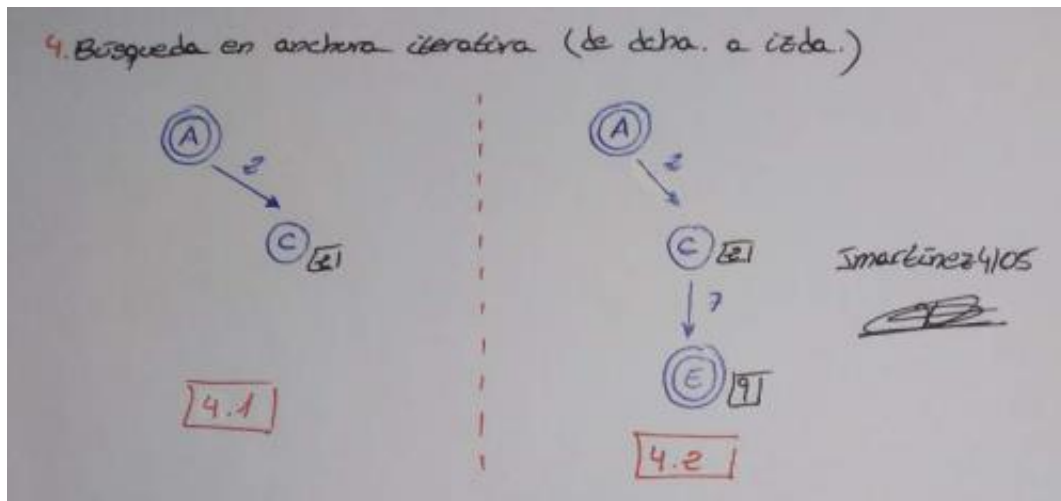


En 3.1 expandimos el nodo inicial.

En 3.2 expandimos B.

En 3.3 expandimos C, continuamos expandiendo el nodo D, pero al no tener hijos no cambia. Por último expandimos en nodo E y llegamos al estado meta. El costo final del camino es de 9.

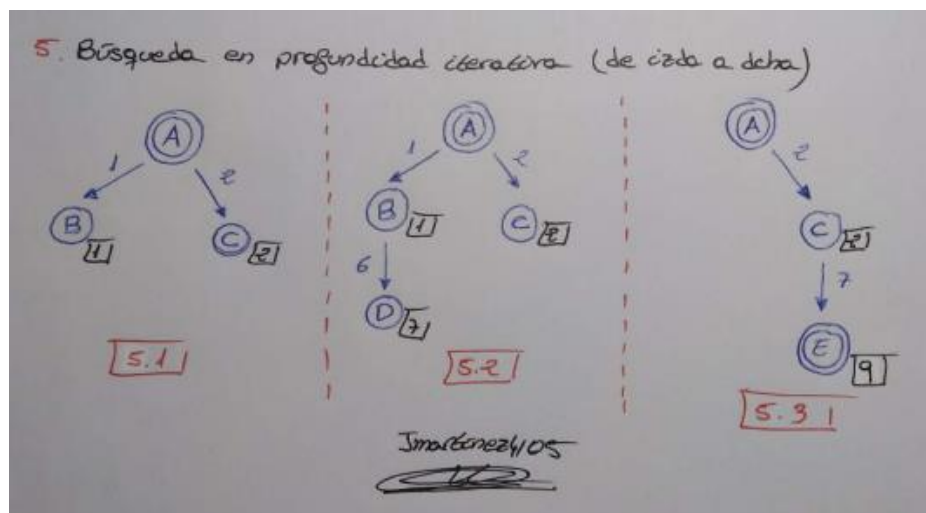
4. Búsqueda en Anchura Iterativa (de derecha a izquierda)



En la primera iteración, se genera como máximo 1 nodo hijo en cada expansión de un nodo padre, como ocurre en 4.1.

En 4.2 seguimos en la primera iteración, pero expandimos el nodo C. Al intentar expandir el nodo E, nos damos cuenta que hemos llegado a un estado meta. El coste del camino hallado es de 9.

5. Búsqueda en Profundidad Iterativa (de izquierda a derecha)



En 5.1 comenzamos expandiendo la raíz con una profundidad límite de 1.

En 5.2 pasamos a la segunda iteración, por lo que la profundidad límite aumenta a 2. En este punto expandimos el nodo B e intentamos expandir el nodo D, pero al no tener hijos aplicamos la función de limpiarABIERTA.

En 5.3 expandimos C y al expandir E llegamos a un nodo meta, por lo que hemos finalizado. El coste de llegar al estado meta es de 9.

Ejercicio 5

Paso 0: El nodo inicial $n1$ es introducido en ABIERTA y en TABLA_A, con lo que tenemos la siguiente situación:

ABIERTA = $\{n1(100 + 0)\}$

Paso 1: Expandimos el nodo $n1$ de ABIERTA. La cual quedará de la siguiente manera:

ABIERTA = $\{\underline{n2(20+50)}, n3(30+50), n4(40+50), n5(200+50), n6(325+0), n7(250+45)\}$

Paso 2: Expandimos el nodo $n2$ por ser el nodo de ABIERTA con menor valor de la evaluación heurística, $f = (g + h)$ (al ser $g = 20$ y $h = 50$).

ABIERTA = $\{\underline{n3(30+50)}, n4(40+50), n5(200+50), n6(325+0), n7(250+45)\}$

Paso 3: Expandimos el nodo $n3$ y elegimos el siguiente nodo más prometedor, que será $n4$.

ABIERTA = $\{\underline{n4(40+50)}, n5(200+50), n6(325+0), n7(250+45)\}$

Paso 4: Expandimos el nodo $n4$, pero no ocurre nada ya que no tiene nodos hijo, por tanto elegiremos el siguiente nodo con mejor heurística, que es $n5$.

ABIERTA = $\{\underline{n5(200+50)}, n6(300+0), n7(250+45)\}$

Debemos observar que el nodo $n6$ ha actualizado su valor a $(300+0)$ ya que el valor al pasar por el nodo expandido $n5$ es menor que su valor anterior $(325 + 0)$, de esta forma $n1$ deja de ser su nodo padre y pasa a serlo $n5$.

Paso 5: Expandimos el nodo $n7$ y por el mismo motivo que antes actualizamos el valor de $n6$ a $(275 + 0)$

ABIERTA = $\{n6(275+0)\}$

Paso 6: Al expandir el nodo $n6$ nos percatamos que hemos llegado al nodo meta y el algoritmo finaliza.

Ejercicio 6

- **Paso 1:** Expandimos el nodo inicial D, generando sus nodos hijos {E(30), F(29)}. Elegimos el nodo F por ser el mejor de todos los nodos hijos. A continuación aceptamos el nodo F, como nuevo nodo actual de sustitución de D, debido a que el valor de la función de evaluación heurística de F, es mejor o igual a la de D ($30 \leq 30$).
- **Paso 2:** Expandimos el nodo actual F, generando sus hijos {C(30), G(29)}. Elegimos el nodo G ya que es el mejor hijo generado por F. Elegimos el nuevo nodo G, como nuevo nodo actual en sustitución a F, debido a que el nuevo valor de la función de evaluación heurística del nodo G es mejor o igual a la del nodo F ($29 \leq 29$).
- **Paso 3:** Expandimos el nodo actual G, generando sus hijos {H(29)}. Elegimos el nodo H ya que es el mejor y único hijo generado por G. Elegimos el nuevo nodo H, como nuevo nodo actual en sustitución a G, debido a que el nuevo valor de la función de evaluación heurística del nodo H es mejor o igual a la del nodo G ($29 \leq 29$).
- **Paso 4:** Expandimos el nodo actual H, generando sus hijos {A(30)}. Elegimos el nodo A ya que es el único generado por H. Rechazamos el nodo A, como nuevo nodo, debido a que el valor de la función heurística de A NO es mejor o igual a la de H. No se cumple que $30 \leq 29$.

La búsqueda termina en este punto y el algoritmo devolverá el nodo H(29) como mejor nodo encontrado. El grafo final quedaría de la siguiente manera:

