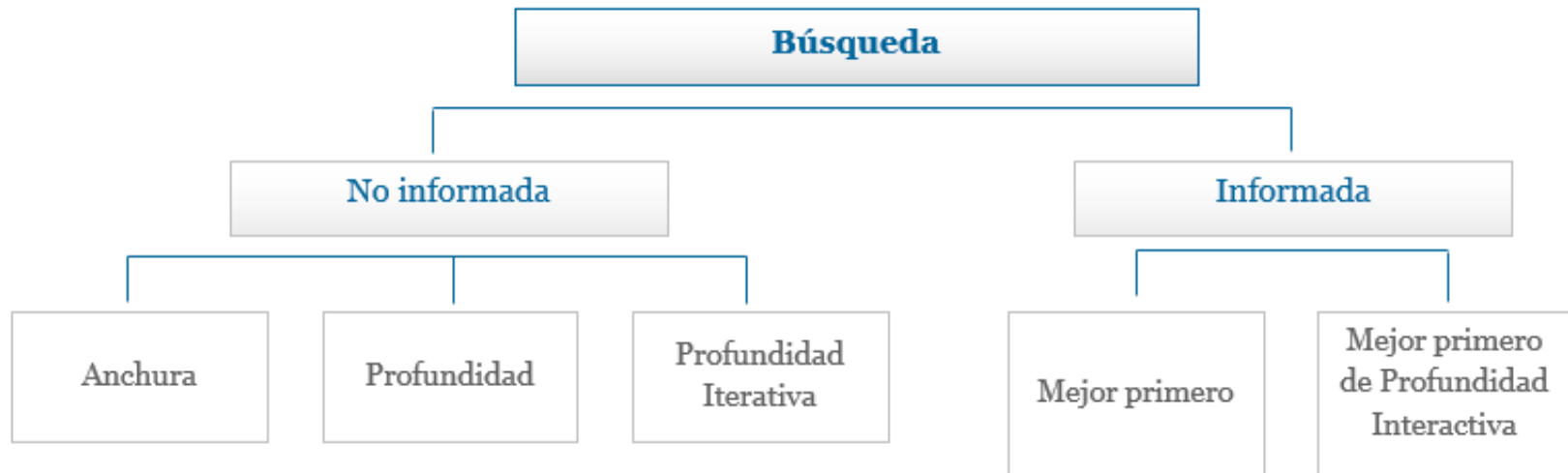


Inteligencia Artificial e Ingeniería del Conocimiento

Elena Verdú Pérez

Búsqueda

¿Cómo estudiar este tema?



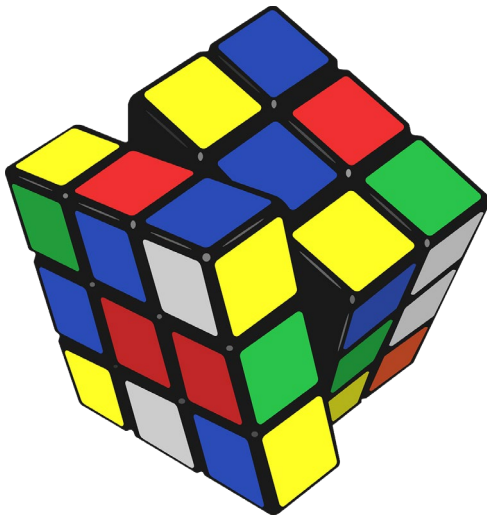
» TEMA 2. BÚSQUEDA

[Esquema Tema]

IDEAS CLAVE	LO + RECOMENDADO	+ INFORMACIÓN	TEST
<p>¿Cómo estudiar este tema?</p> <p>Conceptos comunes de búsqueda</p> <p>Búsqueda no informada</p> <p>Búsqueda informada</p>	<p>Lecciones magistrales</p> <p>TV Búsqueda informada y no informada</p> <p>No dejes de leer...</p> <p>Transparencias elaboradas por Pedro Meseguer</p> <p>No dejes de ver...</p> <p>TV Resolviendo un problema de encontrar caminos en retículas con A*</p>	<p>A fondo</p> <p>Heuristic Search</p> <p>Búsqueda Heurística en OpenCourseWare</p> <p>Encontrar caminos en juegos</p> <p>Bibliografía</p> <p>Recursos externos</p> <p>AIMA</p>	

Conceptos de búsqueda

Modelar un problema de forma que la solución corresponda a un camino que nos lleve desde un estado inicial a un estado meta.



Problema de enrutamiento
de vehículos

Cubo de Rubik

Problema de la mochila

Conceptos de búsqueda

- Un problema de búsqueda se caracteriza por:
 - Los estados que conforman el espacio de búsqueda
 - El estado inicial
 - El test de meta
 - La función de generación de sucesores
- Caso básico:
 - Entorno estático y observable
 - Espacio de estados discreto
 - Información completa
 - Ejecución de acciones determinista

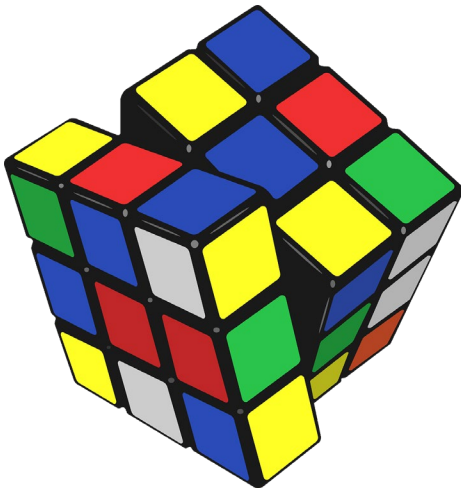


Conceptos de búsqueda

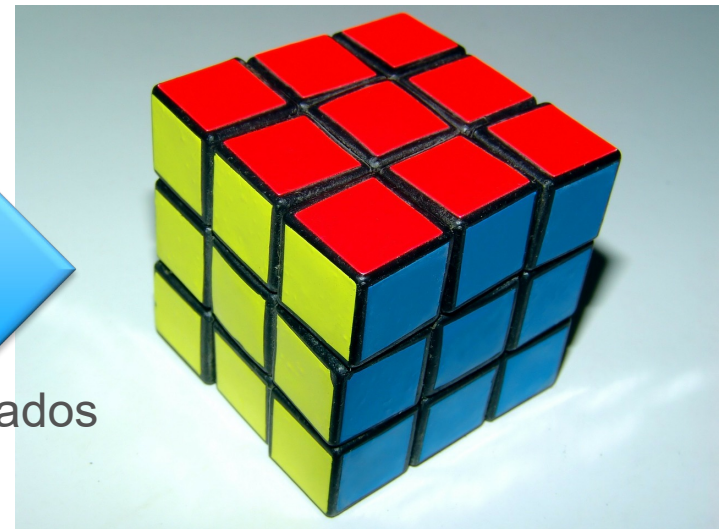
- Cuando se aplica la función de generación de sucesores sobre un estado se dice que **el estado se expande**
- Cuando un estado nuevo aparece como consecuencia de la función de generación de sucesores se dice que el **estado se genera**
- Comúnmente cada estado almacena:
 - Su propia información
 - Estado padre
 - Acción que generó el estado
 - **Coste** de llegar desde el estado inicial hasta dicho estado: $g(s)$

Conceptos de búsqueda

Problema de explosión combinatoria



$4,32 \times 10^{19}$ posibles estados



Ejemplo: el problema de las jarras de agua

Se tienen dos jarras vacías. Una tiene 4 litros de capacidad y otra tiene 3 litros de capacidad.

Ninguna de las jarras tiene marcas de medición y no hay otro dispositivo de medida.

Hay un grifo para llenar las jarras.

¿Cómo podemos conseguir tener 2 litros de agua en la jarra de 4 litros?



Ejemplo: el problema de las jarras de agua

Espacio de estados (x, y) tal que:

- x es el número de litros en la jarra de 4 litros y puede tomar los valores 0, 1, 2, 3, 4
- y es el número de litros en la jarra de 3 litros y puede tomar los valores 0, 1, 2, 3.

Estado inicial: $(0, 0)$

Estado objetivo: $(2, n)$
para cualquier valor de n

Suposiciones:

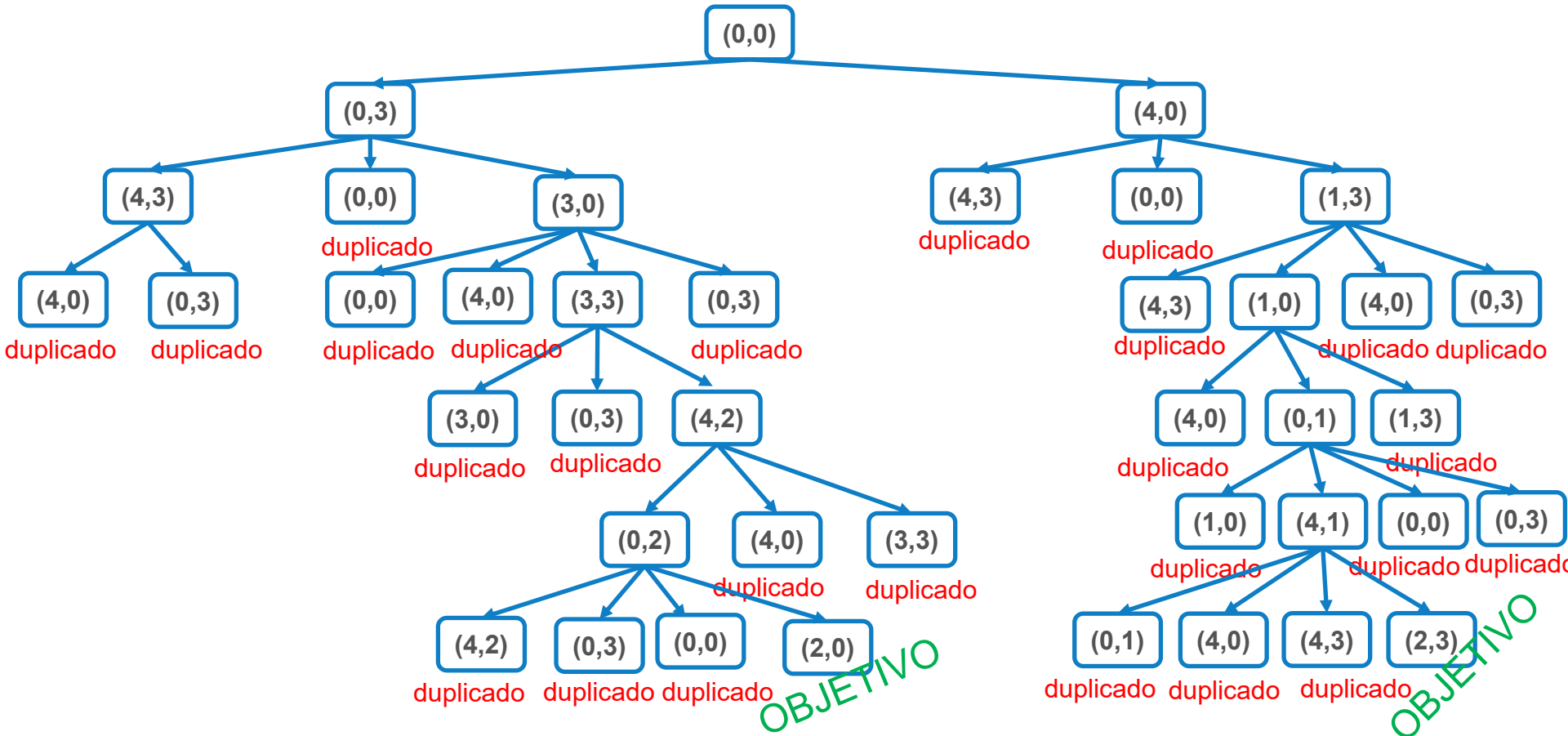
- Las jarras se pueden vaciar en el suelo.
- Se puede verter agua de una jarra a otra.

Ejemplo: el problema de las jarras de agua

Función de generación de sucesores

N	Antecedente	Consecuente	Descripción
1	(x,y) si $x < 4$	$\rightarrow (4,y)$	Llenar la jarra de 4 litros
2	(x,y) si $y < 3$	$\rightarrow (x, 3)$	Llenar la jarra de 3 litros
3	(x,y) si $x > 0$	$\rightarrow (0, y)$	Vaciar la jarra de 4 litros
4	(x,y) si $y > 0$	$\rightarrow (x, 0)$	Vaciar la jarra de 3 litros
5	(x,y) si $x+y \geq 4$ e $y > 0$	$\rightarrow (4, y - (4-x))$	Verter agua desde la jarra de 3 litros a la jarra de 4 litros hasta que se llene
6	(x,y) si $x+y \geq 3$ y $x > 0$	$\rightarrow (x - (3-y), 3)$	Verter agua desde la jarra de 4 litros a la jarra de 3 litros hasta que se llene
7	(x,y) si $x+y \leq 4$ e $y > 0$	$\rightarrow (x+y, 0)$	Verter todo el agua de la jarra de 3 en la jarra de 4
8	(x,y) si $x+y \leq 3$ y $x > 0$	$\rightarrow (0, x+y)$	Verter todo el agua de la jarra de 4 en la jarra de 3

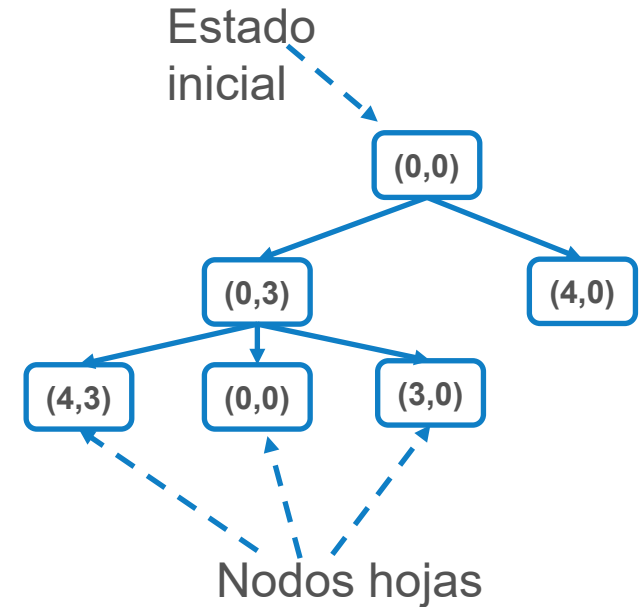
Ejemplo: el problema de las jarras de agua



Exploración del espacio
de búsqueda: ÁRBOL DE
BÚSQUEDA

Conceptos de búsqueda

- Árbol de búsqueda
- Lista abierta de estados no expandidos
- Lista cerrada de estados ya expandidos
- Pueden existir distintos caminos desde un nodo inicial a un estado
→ DETECCIÓN DE DUPLICADOS.
- Un nuevo nodo duplicado se puede descartar si tiene igual o peor coste, o actualizar si tiene mejor coste.



Conceptos de búsqueda

Características fundamentales de los algoritmos:

- 1 **Compleitud:** el algoritmo garantiza que, bajo tiempo y memoria infinitos, devuelve una solución si la hay.
- 2 **Optimalidad:** el algoritmo garantiza que la solución devuelta es óptima, es decir, es una solución de coste mínimo entre el conjunto total de soluciones.
- 3 **Complejidad temporal:** el tiempo que tarda el algoritmo en encontrar una solución en notación O grande en función del tamaño del árbol de búsqueda en el peor de los casos.
- 4 **Complejidad espacial:** la memoria máxima que necesita el algoritmo para encontrar una solución en notación O grande en función del tamaño del árbol de búsqueda en el peor de los casos.

Complejidad, se puede medir en base a:

- **factor de ramificación b** \rightarrow número máximo de acciones aplicables en cualquier estado
- **Profundidad d** \rightarrow longitud mínima de cualquier solución

Búsqueda no informada

Los algoritmos de **búsqueda no informada** o **búsqueda a ciegas** no utilizan información propia del problema.

La estrategia de expansión no discrimina entre estados que puedan ser más o menos prometedores.

1

Búsqueda en anchura

2

Búsqueda en profundidad

3

Búsqueda de profundidad iterativa

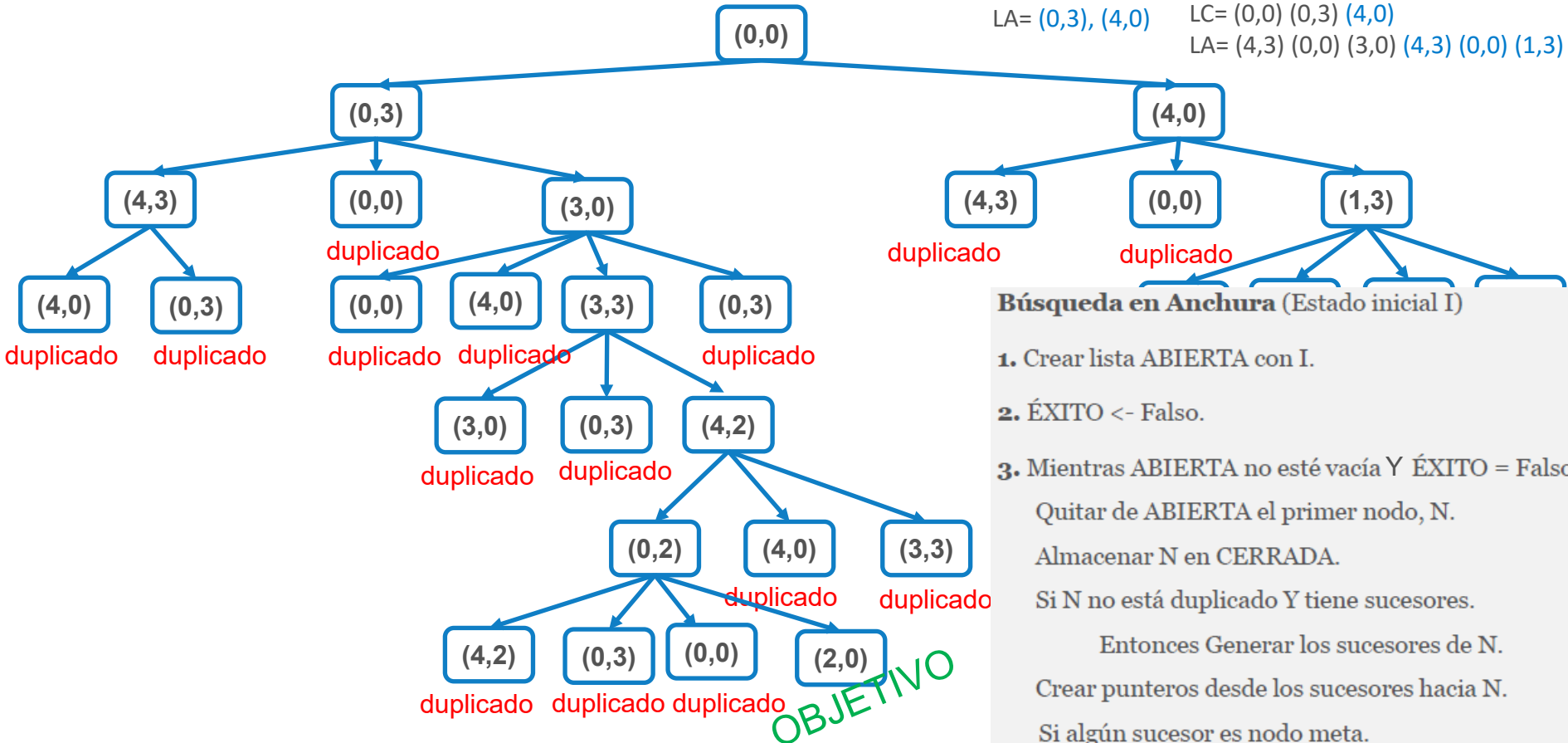
Búsqueda en anchura

- Los nodos se expanden por niveles o capas
- La distancia g desde el estado inicial hasta los nodos de una misma capa es la misma.
- Estrategia FIFO

Búsqueda en anchura

Iteración 1:
LA= (0,0)
Iteración 2:
LC= (0,0)
LA= (0,3), (4,0)

Iteración 3:
LC= (0,0) (0,3)
LA= (4,0) (4,3) (0,0) (3,0)
Iteración 4:
LC= (0,0) (0,3) (4,0)
LA= (4,3) (0,0) (3,0) (4,3) (0,0) (1,3)



Búsqueda en Anchura (Estado inicial I)

1. Crear lista ABIERTA con I.
2. ÉXITO <- Falso.
3. Mientras ABIERTA no esté vacía Y ÉXITO = Falso.
 - Quitar de ABIERTA el primer nodo, N.
 - Almacenar N en CERRADA.
 - Si N no está duplicado Y tiene sucesores.
 - Entonces Generar los sucesores de N.
 - Crear punteros desde los sucesores hacia N.
 - Si algún sucesor es nodo meta.
 - Entonces ÉXITO <- Verdadero.
 - Si no Añadir los sucesores al final de ABIERTA.
4. Si ÉXITO.
 - Entonces devolver el camino inverso de N a I.
 - Si no, devolver fracaso.

Búsqueda completa

Búsqueda óptima

Búsqueda en anchura

Iteración 1:

LA= (0,0)

Iteración 2:

LC= (0,0)

LA= (0,3), (4,0)

Iteración 3:

LC= (0,0) (0,3)

LA= (4,0) (4,3) (0,0) (3,0)

Iteración 4:

LC= (0,0) (0,3) (4,0)

LA= (4,3) (0,0) (3,0) (4,3) (0,0) (1,3)

Iteración 5:

LC= (0,0) (0,3) (4,0) (4,3)

LA= (0,0) (3,0) (4,3) (0,0) (1,3) (4,0) (0,3)

Iteración 6:

LC= (0,0) (0,3) (4,0) (4,3) (0,0)

LA= (3,0) (4,3) (0,0) (1,3) (4,0) (0,3)

Iteración 7:

LC= (0,0) (0,3) (4,0) (4,3) (0,0) (3,0)

LA= (4,3) (0,0) (1,3) (4,0) (0,3) (0,0) (4,0) (3,3) (0,3)

Iteración 8:

LC= (0,0) (0,3) (4,0) (4,3) (0,0) (3,0) (4,3)

LA= (0,0) (1,3) (4,0) (0,3) (0,0) (4,0) (3,3) (0,3)

Iteración 9:

LC= (0,0) (0,3) (4,0) (4,3) (0,0) (3,0) (4,3) (0,0)

LA= (1,3) (4,0) (0,3) (0,0) (4,0) (3,3) (0,3)

Iteración 10

LC= (0,0) (0,3) (4,0) (4,3) (0,0) (3,0) (4,3) (0,0) (1,3)

LA= (4,0) (0,3) (0,0) (4,0) (3,3) (0,3) (4,3) (1,0) (4,0) (0,3)

Iteración 11:

LC= (0,0) (0,3) (4,0) (4,3) (0,0) (3,0) (4,3) (0,0) (1,3) (4,0)

LA= (0,3) (0,0) (4,0) (3,3) (0,3) (4,3) (1,0) (4,0) (0,3)

Iteración 12:

LC= (0,0) (0,3) (4,0) (4,3) (0,0) (3,0) (4,3) (0,0) (1,3) (4,0) (0,3)

LA= (0,0) (4,0) (3,3) (0,3) (4,3) (1,0) (4,0) (0,3)

Iteración 13:

LC= (0,0) (0,3) (4,0) (4,3) (0,0) (3,0) (4,3) (0,0) (1,3) (4,0) (0,3) (0,0)

LA= (4,0) (3,3) (0,3) (4,3) (1,0) (4,0) (0,3)

Iteración 14:

LC= (0,0) (0,3) (4,0) (4,3) (0,0) (3,0) (4,3) (0,0) (1,3) (4,0) (0,3) (0,0) (4,0)

LA= (3,3) (0,3) (4,3) (1,0) (4,0) (0,3)

Iteración 15:

LC= (0,0) (0,3) (4,0) (4,3) (0,0) (3,0) (4,3) (0,0) (1,3) (4,0) (0,3) (0,0) (4,0) (3,3)

LA= (0,3) (4,3) (1,0) (4,0) (0,3) (3,0), (0,3) (4,2)

Iteración 16:

LC= (0,0) (0,3) (4,0) (4,3) (0,0) (3,0) (4,3) (0,0) (1,3) (4,0) (0,3) (0,0) (4,0) (3,3) (0,3)

LA= (4,3) (1,0) (4,0) (0,3) (3,0), (0,3) (4,2)

Iteración 17:

LC= (0,0) (0,3) (4,0) (4,3) (0,0) (3,0) (4,3) (0,0) (1,3) (4,0) (0,3) (0,0) (4,0) (3,3) (0,3) (4,3)

LA= (1,0) (4,0) (0,3) (3,0), (0,3) (4,2)

Búsqueda en anchura

Iteración 18:

LC= (0,0) (0,3) (4,0) (4,3) (0,0) (3,0) (4,3) (0,0) (1,3) (4,0) (0,3) (0,0) (4,0) (3,3) (0,3) (4,3) (1,0)

LA= (4,0) (0,3) (3,0), (0,3) (4,2) (4,0) (0,1) (1,3)

Iteración 19:

LC= (0,0) (0,3) (4,0) (4,3) (0,0) (3,0) (4,3) (0,0) (1,3) (4,0) (0,3) (0,0) (4,0) (3,3) (0,3) (4,3) (1,0) (4,0)

LA= (0,3) (3,0), (0,3) (4,2) (4,0) (0,1) (1,3)

Iteración 20:

LC= (0,0) (0,3) (4,0) (4,3) (0,0) (3,0) (4,3) (0,0) (1,3) (4,0) (0,3) (0,0) (4,0) (3,3) (0,3) (4,3) (1,0) (4,0) (0,3)

LA= (3,0), (0,3) (4,2) (4,0) (0,1) (1,3)

Iteración 21:

LC= (0,0) (0,3) (4,0) (4,3) (0,0) (3,0) (4,3) (0,0) (1,3) (4,0) (0,3) (0,0) (4,0) (3,3) (0,3) (4,3) (1,0) (4,0) (0,3) (3,0)

LA= (0,3) (4,2) (4,0) (0,1) (1,3)

Iteración 22:

LC= (0,0) (0,3) (4,0) (4,3) (0,0) (3,0) (4,3) (0,0) (1,3) (4,0) (0,3) (0,0) (4,0) (3,3) (0,3) (4,3) (1,0) (4,0) (0,3) (3,0) (0,3)

LA= (4,2) (4,0) (0,1) (1,3)

Iteración 23:

LC= (0,0) (0,3) (4,0) (4,3) (0,0) (3,0) (4,3) (0,0) (1,3) (4,0) (0,3) (0,0) (4,0) (3,3) (0,3) (4,3) (1,0) (4,0) (0,3) (3,0) (0,3) (4,2)

LA= (4,0) (0,1) (1,3) (0,2) (4,0) (3,3)

Iteración 24:

LC= (0,0) (0,3) (4,0) (4,3) (0,0) (3,0) (4,3) (0,0) (1,3) (4,0) (0,3) (0,0) (4,0) (3,3) (0,3) (4,3) (1,0) (4,0) (0,3) (3,0) (0,3) (4,2) (4,0)

LA= (0,1) (1,3) (0,2) (4,0) (3,3)

Iteración 25:

LC= (0,0) (0,3) (4,0) (4,3) (0,0) (3,0) (4,3) (0,0) (1,3) (4,0) (0,3) (0,0) (4,0) (3,3) (0,3) (4,3) (1,0) (4,0) (0,3) (3,0) (0,3) (4,2) (4,0) (0,1)

LA= (1,3) (0,2) (4,0) (3,3) (1,0) (4,1) (0,0), (0,3)

Iteración 26:

LC= (0,0) (0,3) (4,0) (4,3) (0,0) (3,0) (4,3) (0,0) (1,3) (4,0) (0,3) (0,0) (4,0) (3,3) (0,3) (4,3) (1,0) (4,0) (0,3) (3,0) (0,3) (4,2) (4,0) (0,1) (1,3)

LA= (0,2) (4,0) (3,3) (1,0) (4,1) (0,0), (0,3)

Iteración 27:

LC= (0,0) (0,3) (4,0) (4,3) (0,0) (3,0) (4,3) (0,0) (1,3) (4,0) (0,3) (0,0) (4,0) (3,3) (0,3) (4,3) (1,0) (4,0) (0,3) (3,0) (0,3) (4,2) (4,0) (0,1) (1,3) (0,2)

LA= (4,0) (3,3) (1,0) (4,1) (0,0), (0,3) (4,2) (0,3) (0,0),(2,0)

Búsqueda en profundidad

- Los nodos se expanden prefiriendo aquellos nodos de mayor profundidad
- El algoritmo desciende por el árbol de búsqueda hasta que encuentra un estado meta o un estado sin sucesores.
- El algoritmo retrocede y busca por los nodos hermanos sin expandir de los nodos ya expandidos.
- Estrategia LIFO

Búsqueda en profundidad

1. Crear lista ABIERTA con I.

2. ÉXITO <- Falso.

3. Mientras ABIERTA no esté vacía Y ÉXITO = Falso.

Quitar de ABIERTA el primer nodo, N.

Añadir N al CAMINO ACTUAL.

Si N tiene sucesores.

Entonces Generar los sucesores de N.

Crear punteros desde los sucesores hacia N.

Si algún sucesor es nodo meta.

Entonces ÉXITO <- Verdadero.

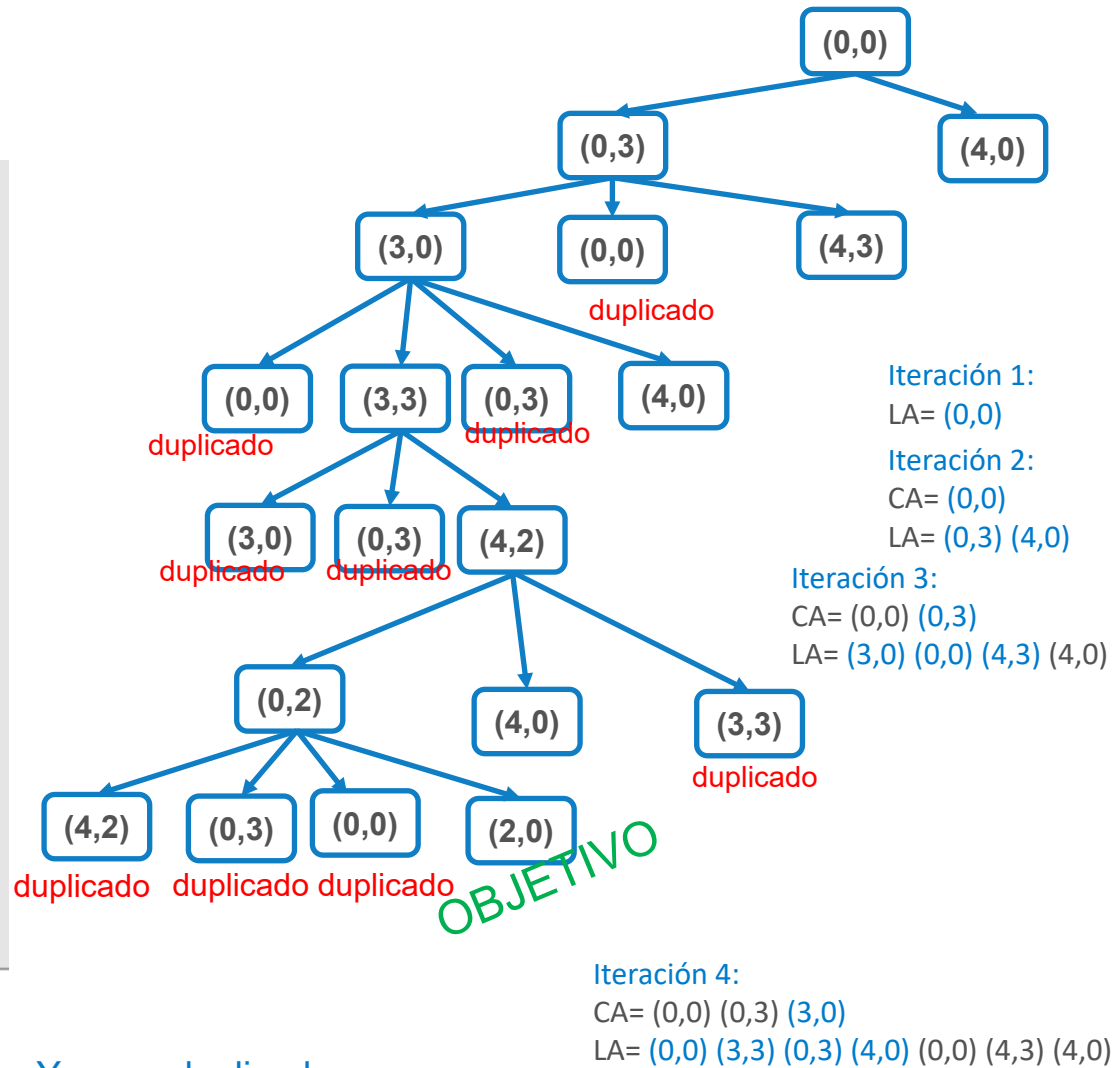
Si no Añadir los sucesores al comienzo de ABIERTA.

Si no Eliminar N del CAMINO ACTUAL.

4. Si ÉXITO.

Entonces devolver el CAMINO ACTUAL.

Si no, devolver fracaso.



Para evitar ciclos: Si N tiene sucesores Y no es duplicado

Búsqueda en profundidad

Iteración 1:

LA= (0,0)

Iteración 2:

CA= (0,0)

LA= (0,3) (4,0)

Iteración 3:

CA= (0,0) (0,3)

LA= (3,0) (0,0) (4,3) (4,0)

Iteración 4:

CA= (0,0) (0,3) (3,0)

LA= (0,0) (3,3) (0,3) (4,0) (0,0) (4,3) (4,0)

Iteración 5:

CA= (0,0) (0,3) (3,0)

LA= (3,3) (0,3) (4,0) (0,0) (4,3) (4,0)

Iteración 6:

CA= (0,0) (0,3) (3,0) (3,3)

LA= (3,0) (0,3) (4,2) (0,3) (4,0) (0,0) (4,3) (4,0)

Iteración 7:

CA= (0,0) (0,3) (3,0) (3,3)

LA= (0,3) (4,2) (0,3) (4,0) (0,0) (4,3) (4,0)

Iteración 8:

CA= (0,0) (0,3) (3,0) (3,3)

LA= (4,2) (0,3) (4,0) (0,0) (4,3) (4,0)

Iteración 9:

CA= (0,0) (0,3) (3,0) (3,3) (4,2)

LA= (0,2) (4,0) (3,3) (0,3) (4,0) (0,0) (4,3) (4,0)

Iteración 10:

CA= (0,0) (0,3) (3,0) (3,3) (4,2) (0,2)

LA= (4,0) (3,3) (0,3) (4,0) (0,0) (4,3) (4,0)

(2,0) es sucesor de (0,2) y es nodo meta.

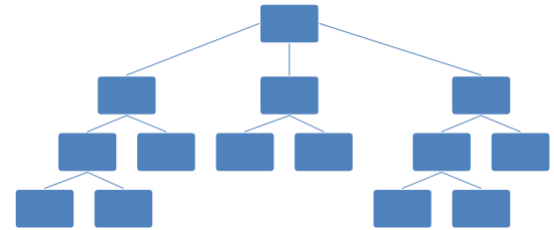
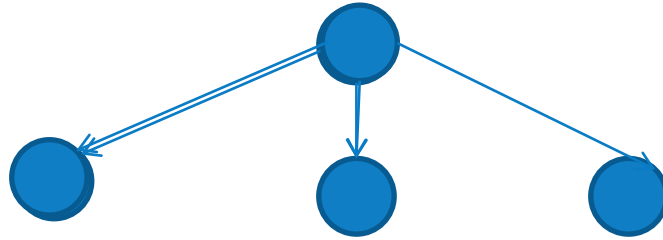
Búsqueda en profundidad

- Diferencias con la búsqueda en anchura
 - Los sucesores se almacenan al comienzo de la lista abierta
 - Se almacena únicamente el camino actual (no hay lista cerrada)
 - La lista abierta contiene los nodos hermanos de los nodos del camino actual que no han sido expandidos
- Incompleta en espacios de búsqueda infinitos.
- Completa en espacio de búsqueda finitos si se evitan ciclos.
- No es óptima.
- La complejidad temporal es del orden de b^d .
- La complejidad espacial es del orden de bd .

Busqueda en profundidad iterativa

- Múltiples búsquedas en profundidad pero acotada.
- La complejidad temporal es del orden de b^d .
- La complejidad espacial es del orden de bd .
- Algoritmo completo y óptimo.

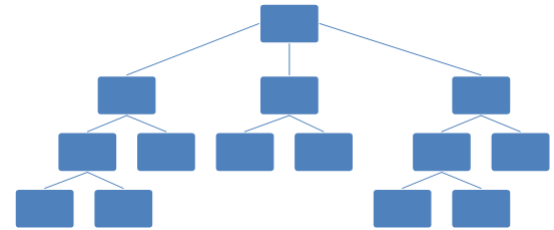
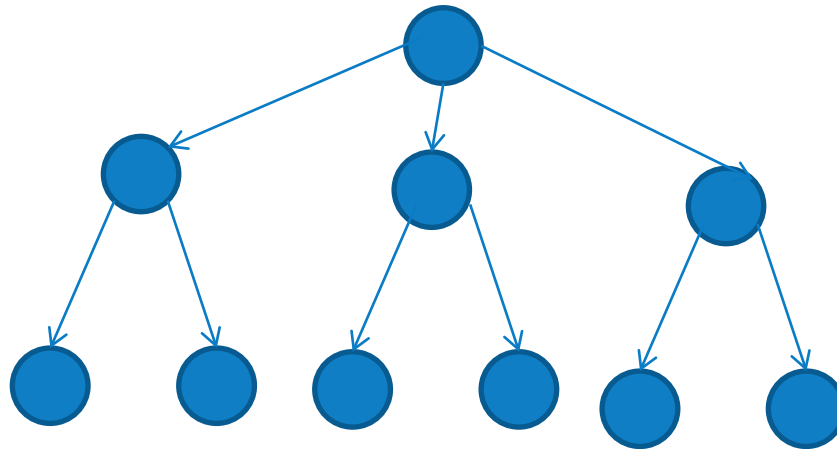
Cota es 0



Busqueda en profundidad iterativa

- Múltiples búsquedas en profundidad pero acotada.
- La complejidad temporal es del orden de b^d .
- La complejidad espacial es del orden de bd .
- Algoritmo completo y óptimo.

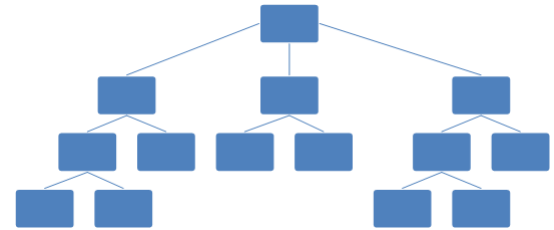
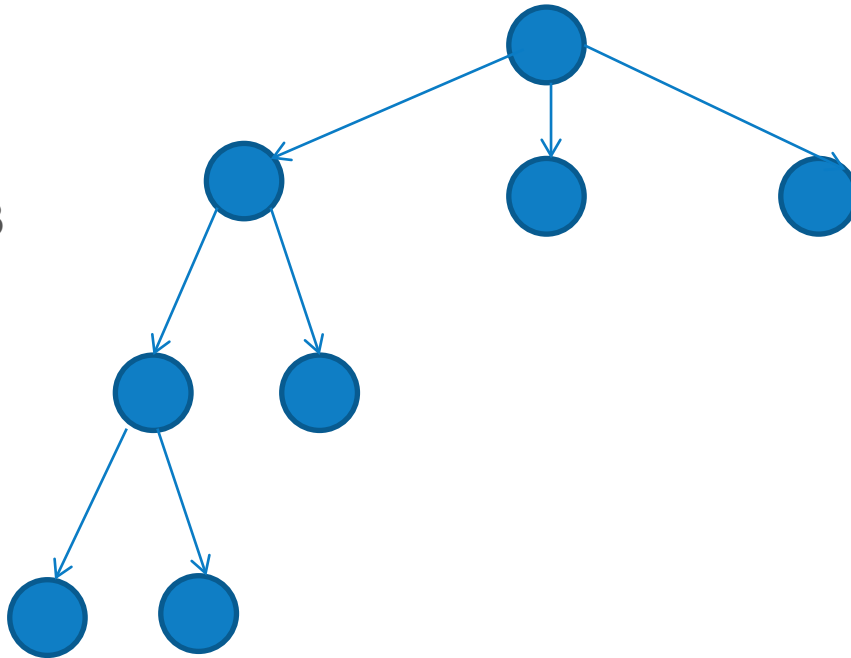
Cota es 2



Busqueda en profundidad iterativa

- Múltiples búsquedas en profundidad pero acotada.
- La complejidad temporal es del orden de b^d .
- La complejidad espacial es del orden de bd .
- Algoritmo completo y óptimo.

Cota es 3



Juego

Plazo: hasta 1 hora antes de la siguiente clase presencial.

Formulario para contestar:

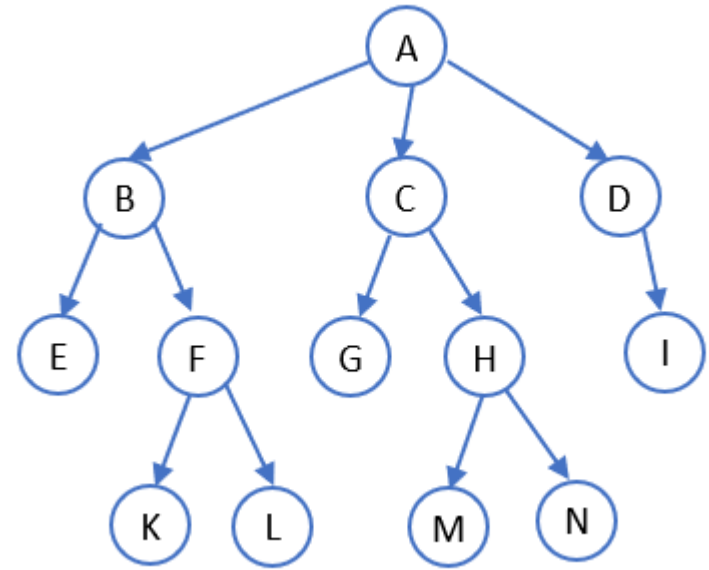
<https://forms.office.com/r/XUvaW2KDyx>

DESAFÍO 1:

Contesta brevemente (no más de 150 palabras) a las siguientes cuestiones: ¿Qué diferencia hay entre la búsqueda no informada y la búsqueda informada? ¿Qué función determina la idoneidad de un estado en búsqueda informada? Sin utilizar una fórmula, describe de qué dos elementos depende esta función.

Juego

DESAFÍOS 2 y 3:



Dado el árbol de la figura donde L es el único nodo meta y A es el nodo inicial, indica:

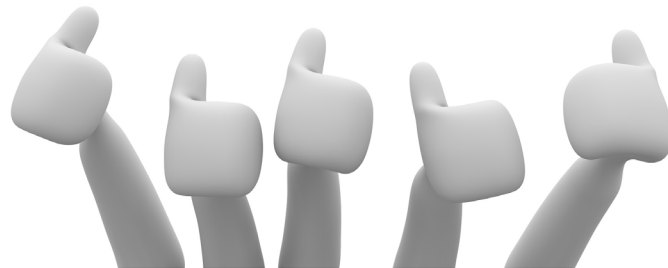
1. El contenido de la lista abierta y de la lista cerrada en cada iteración para el algoritmo de búsqueda en anchura.
2. El contenido de la lista abierta y de la lista con el camino actual en cada iteración para el algoritmo de búsqueda en profundidad.

¿Dudas?



¡Muchas gracias por vuestra atención!

¡Feliz y provechosa semana!





www.unir.net